


```

COUNT(InvoiceDate) AS InvoiceDate_count,
COUNT(UnitPrice) AS UnitPrice_count,
COUNT(CustomerID) AS CustomerID_count,
COUNT(Country) AS Country_count

```

```

FROM `aerial-mission-456103-k9.modulabs_project.modu`

```

작업 정보

결과

자트

JSON

실행 세부정보

실행 그래프

항	InvoiceNo_count	StockCode_count	Description_count	Quantity_count	InvoiceDate_count	UnitPrice_count	CustomerID_count	Country_count
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 ALL**을 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 aerial-mission-456103-k9.modulabs_project.modu

```

```

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 aerial-mission-456103-k9.modulabs_project.modu

```

```

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 aerial-mission-456103-k9.modulabs_project.modu

```

```

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 aerial-mission-456103-k9.modulabs_project.modu

```

```

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 aerial-mission-456103-k9.modulabs_project.modu

```

```

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 aerial-mission-456103-k9.modulabs_project.modu

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 aerial-mission-456103-k9.modulabs_project.modu

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 aerial-mission-456103-k9.modulabs_project.modu;

```

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

결측치 처리 전략

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

```

SELECT DISTINCT Description
FROM aerial-mission-456103-k9.modulabs_project.modu
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 aerial-mission-456103-k9.modulabs_project.modu
WHERE Description is NULL

```

작업 정보
결과
실행 세부정보
실행 그래프

i 이 문으로 modu의 행 1,454개가 삭제되었습니다.
테이블로 이동

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,
  UnitPrice,
  CustomerID,
  Country,
  COUNT(*) AS row_count
FROM aerial-mission-456103-k9.modulabs_project.modu
GROUP BY
  InvoiceNo,
  StockCode,
  Description,
  Quantity,
  InvoiceDate,
  UnitPrice,
  CustomerID,
  Country
HAVING COUNT(*) > 1;
```

작업 정보	결과	실행 세부정보	실행 그래프
1	537634	22520	CHLZS GARDEN TROMEL BLUE
2	539045	22951	60 CAKE CAKES DOLLY GIRL D.
3	539056	22518	CHLZS GARDEN BRUSH BLUE
4	541494	22489	PAUL OF 15 TRADITIONAL DR.
5	541975	22482	MEASURING TAPE BABUSHKA ..
6	541975	20682	RED RETROSPOT CHILDRENS ..
7	541975	20749	ASSORTED COLOUR MINI CARL
8	542967	22131	FOOD CONTAINER SET TULINE
9	543099	21121	SET10 RED POLKA DOT PART..
10	547224	22345	CHRISTMAS METAL TAGS ASS.
11	548804	22483	MEASURING TAPE BABUSHKA ..
12	549537	22865	EASTER TIN CHICKS PINK DOL

중복값 처리

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

```
CREATE OR REPLACE TABLE aerial-mission-456103-k9.modulabs_project.modu AS
SELECT DISTINCT *
FROM aerial-mission-456103-k9.modulabs_project.modu;
```

작업 정보 결과 실행 세부정보 실행 그래프

이 문으로 이름이 modu인 테이블이 교체되었습니다.

행	remaining_rows
1	535187

11-6. 데이터 전처리(3): 오류값 처리

InvoiceNo 살펴보기

- 고유(unique)한 InvoiceNo의 개수를 출력하기

```
SELECT COUNT(DISTINCT InvoiceNo)
FROM `aerial-mission-456103-k9.modulabs_project.modu`
```

행	f0_
1	24446

- 고유한 InvoiceNo를 앞에서부터 100개를 출력하기

```
SELECT DISTINCT InvoiceNo
FROM `aerial-mission-456103-k9.modulabs_project.modu`
LIMIT 100
```

행	InvoiceNo
1	536544
2	536555
3	536558
4	536565
5	536592
6	536596
7	536640
8	536755
9	536756

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

```
SELECT InvoiceNo
FROM `aerial-mission-456103-k9.modulabs_project.modu`
WHERE InvoiceNo like 'C%'
LIMIT 100;
```

작업 정보	결과	차트	JSON	실행 세부정보	실행 그래프
행	InvoiceNo				
1	C537251				
2	C537251				
3	C537251				
4	C537251				
5	C537251				
6	C537251				
7	C537251				
8	C537251				
9	C537251				

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

```
SELECT
ROUND(
COUNTIF(InvoiceNo LIKE 'C%') * 100.0 / COUNT(*),
1
) AS canceled_ratio_percent
FROM aerial-mission-456103-k9.modulabs_project.modu;
```

행	canceled_ratio_percent
1	1.7

StockCode 살펴보기

- 고유한 **StockCode** 의 개수를 출력하기

```
Select count(distinct StockCode)
FROM `aerial-mission-456103-k9.modulabs_project.modu`
```

행	f0_
1	3958

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

```
SELECT StockCode, COUNT(*) AS sell_cnt
FROM project_name.modulabs_project.data
GROUP BY StockCode
ORDER BY sell_cnt DESC
LIMIT 10
```

행	StockCode	sell_cnt
1	85123A	2301
2	22423	2192
3	85099B	2156
4	47566	1720
5	20725	1626
6	84879	1489
7	22197	1468
8	22720	1465
9	21212	1367
10	22383	1328

- StockCode** 의 컬럼에 있던 값 중에서 숫자를 제외한 문자만 남기고 문자가 몇 자리 수 인지 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 `aerial-mission-456103-k9.modulabs_project.modu`
)
WHERE number_count <= 1

```

행	StockCode	number_count
1	DOT	0
2	M	0
3	AMAZONFEE	0
4	C2	1
5	m	0
6	POST	0
7	DCGSSGIRL	0
8	DCGSSBOY	0
9	S	0
10	B	0
11	BANK CHARGES	0
12	D	0
13	PADS	0
14	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 퍼센트인지 구하기 (소수점 두 번째 자리까지)

```

SELECT
  ROUND(
    COUNTIF(number_count <= 1) * 100.0 / COUNT(*),
    2
  ) AS percentage_of_low_digit_stockcodes
FROM (
  SELECT
    StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM `aerial-mission-456103-k9.modulabs_project.modu`
);

```

행	percentage_of_low
1	0.55

- 제품과 관련되지 않은 거래 기록을 제거하기

```

DELETE FROM `aerial-mission-456103-k9.modulabs_project.modu`
WHERE StockCode IN (
  SELECT StockCode
  FROM (
    SELECT DISTINCT StockCode,
      LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
    FROM `aerial-mission-456103-k9.modulabs_project.modu`
  )
)

```

```

)
WHERE number_count <= 1
);

```

이 문으로 modu의 행 2,919개가 삭제되었습니다.

Description 살펴보기

- 고유한 Description 별 출현 빈도를 계산하고 상위 30개를 출력하기

```

SELECT Description, COUNT(*) AS description_cnt
FROM `aerial-mission-456103-k9.modulabs_project.modu`
WHERE Description IS NOT NULL
GROUP BY Description
ORDER BY description_cnt DESC
LIMIT 30;

```

행	Description	description_cnt
1	WHITE HANGING HEART T-LIG...	2353
2	REGENCY CAKESTAND 3 TIER	2187
3	JUMBO BAG RED RETROSPOT	2153
4	PARTY BUNTING	1719
5	LUNCH BAG RED RETROSPOT	1625
6	ASSORTED COLOUR BIRD ORN...	1488
7	SET OF 3 CAKE TINS PANTRY ...	1465
8	PACK OF 72 RETROSPOT CAK...	1367
9	LUNCH BAG BLACK SKULL	1323
10	NATURAL SLATE HEART CHAL...	1272
11	JUMBO BAG PINK POLKADOT	1245
12	HEART OF WICKER SMALL	1230
13	JAM MAKING SET WITH JARS	1220
14	JUMBO STORAGE BAG SUKI	1211
15	PAPER CHAIN KIT 50'S CHRIST...	1194
16	JUMBO SHOPPER VINTAGE RE...	1192
17	LUNCH BAG CARS BLUE	1185
18	LUNCH BAG SPACEBOY DESIGN	1177
19	JAM MAKING SET PRINTED	1177
20	RECIPE BOX PANTRY YELLOW ...	1166

- 서비스 관련 정보를 포함하는 행들을 제거하기

```

DELETE
FROM `aerial-mission-456103-k9.modulabs_project.modu`
WHERE REGEXP_CONTAINS(UPPER(Description), r'NEXT DAY CARRIAGE|HIGH RESOLUTION IMAGE|MANUAL|ADJUST|SAMPLE|TE

```

이 문으로 modu의 행 473개가 삭제되었습니다.

- 대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기


```
CREATE OR REPLACE TABLE `aerial-mission-456103-k9.modulabs_project.modu` AS
SELECT
  * EXCEPT (Description),
  UPPER(Description) AS Description
FROM `aerial-mission-456103-k9.modulabs_project.modu`
```

이 문으로 이름이 modu인 테이블이 교체되었습니다.

UnitPrice 살펴보기

- UnitPrice의 최소값, 최대값, 평균을 구하기

```
SELECT # [[YOUR QUERY]] AS min_price, # [[YOUR QUERY]] AS max_price, # [[YOUR QUERY]] AS avg_price
FROM project_name.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

- 단가가 0원인 거래의 개수, 구매 수량(Quantity)의 최소값, 최대값, 평균 구하기

```
SELECT # [[YOUR QUERY]] AS cnt_quantity, # [[YOUR QUERY]] AS min_quantity, # [[YOUR QUERY]] AS max_quantity, # [[YOUR QU
FROM project_name.modulabs_project.data
WHERE # [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

- UnitPrice = 0를 제거하고 일관된 데이터셋을 유지하기

```
CREATE OR REPLACE TABLE project_name.modulabs_project.data AS
SELECT *
FROM project_name.modulabs_project.data
WHERE # [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

11-7. RFM 스코어

Recency

- InvoiceDate 컬럼을 연월일 자료형으로 변경하기

```
SELECT # [[YOUR QUERY]] AS InvoiceDay, *
FROM project_name.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

- 가장 최근 구매 일자를 MAX() 함수로 찾아보기

```
SELECT
  # [[YOUR QUERY]] AS most_recent_date,
  # [[YOUR QUERY]] AS InvoiceDay,
  *
FROM project_name.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

- 유저 별로 가장 큰 InvoiceDay를 찾아서 가장 최근 구매일로 저장하기

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

[결과 이미지를 넣어주세요]

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

```
SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM project_name.modulabs_project.data
  GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]

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

```
CREATE OR REPLACE TABLE project_name.modulabs_project.user_r AS
# [[YOUR QUERY]]
```

[결과 이미지를 넣어주세요]

Frequency

- 고객마다 고유한 InvoiceNo의 수를 세어보기

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

[결과 이미지를 넣어주세요]

- 각 고객 별로 구매한 아이템의 총 수량 더하기

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

[결과 이미지를 넣어주세요]

- 전체 거래 건수 계산과 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]

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]

- 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_name.modulabs_project.user_rfm AS
SELECT
  rf.CustomerID AS CustomerID,
  rf.purchase_cnt,
  rf.item_cnt,
  rf.recency,
  ut.user_total,
  # [[YOUR QUERY]] AS user_average
FROM project_name.modulabs_project.user_rf rf
LEFT JOIN (
  -- 고객 별 총 지출액
  SELECT
    # [[YOUR QUERY]]
  ) ut
ON rf.CustomerID = ut.CustomerID;
```

[결과 이미지를 넣어주세요]

RFM 통합 테이블 출력하기

- 최종 `user_rfm` 테이블을 출력하기

```
# [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

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]

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_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]

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 project_name.modulabs_project.user_data AS

WITH TransactionInfo AS (
  SELECT
    CustomerID,
    # [[YOUR QUERY]] AS total_transactions,
    # [[YOUR QUERY]] AS cancel_frequency
  FROM project_name.modulabs_project.data
  # [[YOUR QUERY]]
)

SELECT u.*, t.* EXCEPT(CustomerID), # [[YOUR QUERY]] AS cancel_rate
FROM `project_name.modulabs_project.user_data` AS u
```

```
LEFT JOIN TransactionInfo AS t
ON # [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

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

```
# [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

회고

[회고 내용을 작성해주세요]

Keep :

Problem :

Try :