

고객을 세그멘테이션하자 [프로젝트]

11-2. 데이터 불러오기

데이터 살펴보기

- 테이블에 있는 10개의 행만 출력하기

```
SELECT *  
FROM hopeful-object-439400-e2.modulabs_project.data  
LIMIT 10;
```

[결과 이미지를 넣어주세요]

| 쿼리 결과 | | | | | | | | | |
|-------|-----------|-----------|-------------|----------|-------------------------|-----------|------------|----------------|--|
| 작업 정보 | | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프 | | | |
| 행 | InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country | |
| 1 | 536414 | 22139 | null | 56 | 2010-12-01 11:52:00 UTC | 0.0 | null | United Kingdom | |
| 2 | 536545 | 21134 | null | 1 | 2010-12-01 14:32:00 UTC | 0.0 | null | United Kingdom | |
| 3 | 536546 | 22145 | null | 1 | 2010-12-01 14:33:00 UTC | 0.0 | null | United Kingdom | |
| 4 | 536547 | 37509 | null | 1 | 2010-12-01 14:33:00 UTC | 0.0 | null | United Kingdom | |
| 5 | 536549 | 85226A | null | 1 | 2010-12-01 14:34:00 UTC | 0.0 | null | United Kingdom | |
| 6 | 536550 | 85044 | null | 1 | 2010-12-01 14:34:00 UTC | 0.0 | null | United Kingdom | |
| 7 | 536552 | 20950 | null | 1 | 2010-12-01 14:34:00 UTC | 0.0 | null | United Kingdom | |
| 8 | 536553 | 37461 | null | 3 | 2010-12-01 14:35:00 UTC | 0.0 | null | United Kingdom | |
| 9 | 536554 | 84670 | null | 23 | 2010-12-01 14:35:00 UTC | 0.0 | null | United Kingdom | |
| 10 | 536589 | 21777 | null | -10 | 2010-12-01 16:50:00 UTC | 0.0 | null | United Kingdom | |

- 전체 데이터는 몇 행으로 구성되어 있는지 확인하기

```
SELECT COUNT(*)  
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | |
|---------|---------|
| 작업 정보 | |
| 결과 | 차트 |
| JSON | 실행 세부정보 |
| 실행 그래프 | |
| 행 | fo_ |
| 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 hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | | | | | | |
|---------|-----------------|-----------------|-------------------|----------------|-------------------|-----------------|------------------|---------------|
| 작업 정보 | | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프 | | |
| 행 | COUNT_InvoiceNo | COUNT_StockCode | COUNT_Description | COUNT_Quantity | COUNT_InvoiceDate | COUNT_UnitPrice | COUNT_CustomerID | COUNT_Country |
| 1 | 541909 | 541909 | 540455 | 541909 | 541909 | 541909 | 406829 | 541909 |

11-4. 데이터 전처리 방법(1): 결측치 제거

컬럼 별 누락된 값의 비율 계산

- 각 컬럼 별 누락된 값의 비율을 계산
 - 각 컬럼에 대해서 누락 값을 계산한 후, 계산된 누락 값을 UNION ALL을 통해 합치기

```
SELECT
    'InvoiceNo' AS column_name,
    ROUND(SUM(CASE WHEN InvoiceNo IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'StockCode' AS column_name,
    ROUND(SUM(CASE WHEN StockCode IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'Description' AS column_name,
    ROUND(SUM(CASE WHEN Description IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'Quantity' AS column_name,
    ROUND(SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'InvoiceDate' AS column_name,
    ROUND(SUM(CASE WHEN InvoiceDate IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'UnitPrice' AS column_name,
    ROUND(SUM(CASE WHEN UnitPrice IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'CustomerID' AS column_name,
    ROUND(SUM(CASE WHEN CustomerID IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data

UNION ALL

SELECT
    'Country' AS column_name,
    ROUND(SUM(CASE WHEN Country IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percent
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | |
|---------|-------------|--------------------|----|
| 작업 정보 | | 결과 | 차트 |
| JSON | | 실행 세부정보 | |
| 행 | column_name | missing_percentage | |
| 1 | InvoiceDate | 0.0 | |
| 2 | CustomerID | 24.93 | |
| 3 | StockCode | 0.0 | |
| 4 | UnitPrice | 0.0 | |
| 5 | Quantity | 0.0 | |
| 6 | Description | 0.27 | |
| 7 | Country | 0.0 | |
| 8 | InvoiceNo | 0.0 | |

결측치 처리 전략

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

```
SELECT DISTINCT Description
FROM hopeful-object-439400-e2.modulabs_project.data
WHERE StockCode = '85123A'
GROUP BY StockCode,Description;
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | |
|---------|------------------------------|---------|--------|
| 작업 정보 | | 결과 | 차트 |
| JSON | | 실행 세부정보 | 실행 그래프 |
| 행 | Description | | |
| 1 | ? | | |
| 2 | wrongly marked carton 22804 | | |
| 3 | CREAM HANGING HEART T-LIG... | | |
| 4 | WHITE HANGING HEART T-LIG... | | |

결측치 처리

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

```
DELETE FROM hopeful-object-439400-e2.modulabs_project.data
WHERE CustomerID IS NULL
OR Description IS NULL
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | |
|---|----|---------|--------|
| 작업 정보 | 결과 | 실행 세부정보 | 실행 그래프 |
| <div><div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><div></div></div></div> <div><div><div></div></div><div><</div></div> | | | |

11-5. 데이터 전처리(2): 중복값 처리

중복값 확인

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

```
SELECT *,
FROM hopeful-object-439400-e2.modulabs_project.data
GROUP BY InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country
HAVING COUNT(*) > 1;
```

[결과 이미지를 넣어주세요]

| 행 | InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country |
|---|-----------|-----------|-----------------------------|----------|-------------------------|-----------|------------|----------------|
| 1 | 557305 | 22545 | CERAMIC HEART FAIRY CAKE .. | 4 | 2011-06-19 14:42:00 UTC | 1.45 | 13568 | United Kingdom |
| 2 | 569943 | 20972 | PINK CREAM FELT CRAFT TRL.. | 1 | 2011-10-06 18:08:00 UTC | 1.25 | 14592 | United Kingdom |
| 3 | 571241 | 72816 | SET/3 CHRISTMAS DECORPAG.. | 1 | 2011-10-14 14:58:00 UTC | 0.95 | 14592 | United Kingdom |
| 4 | 571241 | 22630 | DOLLY GIRL LUNCH BOX | 1 | 2011-10-14 14:58:00 UTC | 1.95 | 14592 | United Kingdom |
| 5 | 571241 | 22807 | SET OF 6 TLIGHTS TOASTTOO.. | 1 | 2011-10-14 14:58:00 UTC | 2.95 | 14592 | United Kingdom |
| 6 | 571241 | 22095 | LADIS ONLY TISSUE BOX | 3 | 2011-10-14 14:58:00 UTC | 0.39 | 14592 | United Kingdom |

중복값 처리

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

```
CREATE OR REPLACE TABLE hopeful-object-439400-e2.modulabs_project.data AS
SELECT DISTINCT *
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| 행 | InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country |
|---|-----------|-----------|-------------|----------|-------------|-----------|------------|---------|
| 1 | 401604 | | | | | | | |

11-6. 데이터 전처리(3): 오류값 처리

InvoiceNo 살펴보기

- 고유(unique)한 InvoiceNo의 개수를 출력하기

```
SELECT COUNT(DISTINCT InvoiceNo)
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| 쿼리 결과 | | | 결과 저장 | |
|-------|-------|--|-------|----|
| 작업 정보 | | | 결과 | 차트 |
| 행 | f0_ | | | |
| 1 | 22190 | | | |

- 고유한 InvoiceNo 를 앞에서부터 100개를 출력하기

```
SELECT DISTINCT InvoiceNo
FROM hopeful-object-439400-e2.modulabs_project.data
LIMIT 100;
```

[결과 이미지를 넣어주세요]

| 쿼리 결과 | | | 결과 저장 | |
|-------|-----------|--|-------|----|
| 작업 정보 | | | 결과 | 차트 |
| 행 | InvoiceNo | | | |
| 1 | 574301 | | | |
| 2 | C575531 | | | |
| 3 | 557305 | | | |
| 4 | 543008 | | | |
| 5 | 549735 | | | |

페이지당 결과 수: 50 1 - 50 (전체 100행)

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

```
SELECT *
FROM hopeful-object-439400-e2.modulabs_project.data
WHERE InvoiceNo LIKE 'C%'
LIMIT 100;
```

[결과 이미지를 넣어주세요]

| 쿼리 결과 | | | 결과 저장 | 데이터 탐색 |
|-------|-----------|-----------|-------------------------------|----------|
| 작업 정보 | | | 결과 | 차트 |
| 행 | InvoiceNo | StockCode | Description | Quantity |
| 1 | C575531 | 22960 | JAM MAKING SET WITH JARS | -4 |
| 2 | C568060 | 22840 | ROUND CAKE TIN VINTAGE RED | -1 |
| 3 | C568060 | 22847 | BREAD BIN DINER STYLE IVORY | -1 |
| 4 | C564983 | 47990A | BLUE HAPPY BIRTHDAY BUNTL... | -20 |
| 5 | C564983 | 47990B | PINK HAPPY BIRTHDAY BUNTL... | -20 |
| 6 | C593709 | 84978 | HANGING HEART JAR T LIGHT ... | -1 |
| 7 | C593709 | 21485 | RETROSPOT HEART HOT WAT... | -1 |
| 8 | C593709 | 22832 | BROCANTE SHELF WITH HOOKS | -2 |
| 9 | C543620 | 21217 | RED RETROSPOT ROUND CAK... | -1 |

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 hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | 결과 저장 | 데이터 |
|---------|-----|--|---------|--------|
| 작업 정보 | | | 결과 | 차트 |
| JSON | | | 실행 세부정보 | 실행 그래프 |
| 행 | f0_ | | | |
| 1 | 2.2 | | | |

StockCode 살펴보기

- 고유한 StockCode 의 개수를 출력하기

```
SELECT COUNT (DISTINCT StockCode)
FROM hopeful-object-439400-e2.modulabs_project.data
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | 결과 저장 | 데이터 탐색 |
|---------|------|--|---------|--------|
| 작업 정보 | | | 결과 | 차트 |
| JSON | | | 실행 세부정보 | 실행 그래프 |
| 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 hopeful-object-439400-e2.modulabs_project.data
GROUP BY StockCode
ORDER BY sell_cnt DESC
LIMIT 10;
```

[결과 이미지를 넣어주세요]

| ← 쿼리 결과 | | | 결과 저장 | 데이터 |
|---------|-----------|----------|---------|--------|
| 작업 정보 | | | 결과 | 차트 |
| JSON | | | 실행 세부정보 | 실행 그래프 |
| 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 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 hopeful-object-439400-e2.modulabs_project.data
)
WHERE number_count = 0 OR number_count =1
```

[결과 이미지를 넣어주세요]

← 쿼리 결과

| 작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 |
|-------|--------------|--------------|------|---------|
| 행 | StockCode | number_count | | |
| 1 | POST | 0 | | |
| 2 | M | 0 | | |
| 3 | PADS | 0 | | |
| 4 | D | 0 | | |
| 5 | BANK CHARGES | 0 | | |
| 6 | DOT | 0 | | |
| 7 | CRUK | 0 | | |
| 8 | C2 | 1 | | |

- **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(SUM(CASE WHEN number_count = 0 OR number_count = 1 THEN 1 ELSE 0 END)/ COUNT(*) * 100 ,
FROM (
  SELECT StockCode,
         LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM hopeful-object-439400-e2.modulabs_project.data
```

[결과 이미지를 넣어주세요]

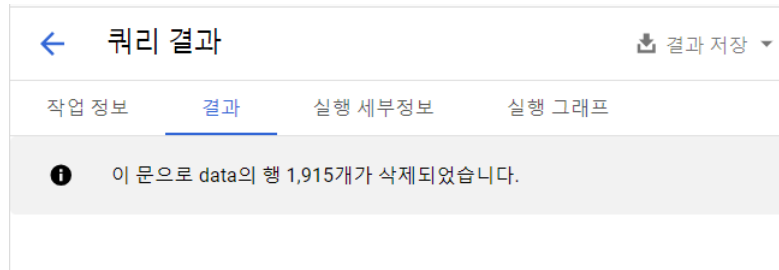
← 쿼리 결과

| 작업 정보 | 결과 | 차트 |
|-------|------|----|
| 행 | f0_ | |
| 1 | 0.48 | |

- 제품과 관련되지 않은 거래 기록을 제거하기

```
DELETE FROM hopeful-object-439400-e2.modulabs_project.data
WHERE StockCode IN (
  SELECT StockCode
  FROM (
    SELECT StockCode,
           LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
    FROM hopeful-object-439400-e2.modulabs_project.data
  )WHERE number_count = 0 OR number_count =1
);
```

[결과 이미지를 넣어주세요]



Description 살펴보기

- 고유한 Description 별 출현 빈도를 계산하고 상위 30개를 출력하기

```
SELECT Description, COUNT(*) AS description_cnt
FROM hopeful-object-439400-e2.modulabs_project.data
GROUP BY Description
LIMIT 30;
```

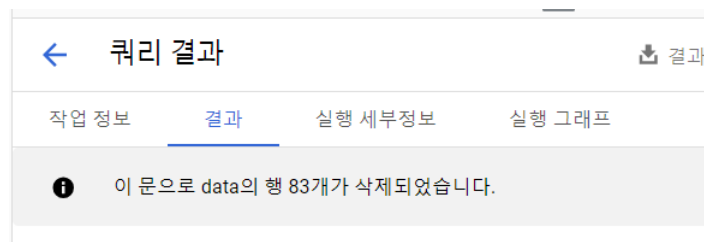
[결과 이미지를 넣어주세요]

| 쿼리 결과 | | | |
|---------------------------------|------------------------------------|-----------------|--|
| 작업 정보 결과 차트 JSON 실행 세부정보 실행 그래프 | | | |
| 행 | Description | description_cnt | |
| 1 | FELTCRAFT PRINCESS OLIVIA DOLL | 254 | |
| 2 | PAPER CHAIN KIT 50'S CHRISTMAS | 1013 | |
| 3 | CHRISTMAS CRAFT LITTLE FRIENDS | 433 | |
| 4 | SET OF 6 RIBBONS VINTAGE CHRISTMAS | 343 | |
| 5 | TRADITIONAL KNITTING NANCY | 523 | |
| 6 | SCANDINAVIAN REDS RIBBONS | 343 | |
| 7 | EMBROIDERED RIBBON REEL EMILY | 87 | |
| 8 | FELTCRAFT PRINCESS LOLA DOLL | 383 | |
| 9 | EMBROIDERED RIBBON REEL SALLY | 63 | |
| 10 | JAM MAKING SET WITH JARS | 966 | |
| 11 | TRADITIONAL CHRISTMAS RIBBONS | 272 | |
| 12 | EMBROIDERED RIBBON REEL ROSIE | 49 | |
| 13 | PINK BLUE FELT CRAFT TRINKET BOX | 465 | |
| 14 | PAPER CHAIN KIT VINTAGE CHRISTMAS | 718 | |

- 서비스 관련 정보를 포함하는 행들을 제거하기

```
DELETE
FROM hopeful-object-439400-e2.modulabs_project.data
WHERE Description IN ('Next Day Carriage', 'High Resolution Image');
```

[결과 이미지를 넣어주세요]



- 대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기

```
CREATE OR REPLACE TABLE hopeful-object-439400-e2.modulabs_project.data AS
SELECT
  * EXCEPT (Description),
```



```
UPPER (Description) AS Description
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

| 쿼리 결과 | | 결과 저장 | 데이터 |
|-------------------------------|----|---------|--------|
| 작업 정보 | 결과 | 실행 세부정보 | 실행 그래프 |
| 이 문으로 이름이 data인 테이블이 교체되었습니다. | | | |


UnitPrice 살펴보기

- UnitPrice의 최소값, 최대값, 평균을 구하기

```
SELECT MIN(UnitPrice) AS min_price, MAX(UnitPrice) AS max_price, AVG(UnitPrice) AS avg_price
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

쿼리 결과

 결과 저장

작업 정보

결과

차트

JSON

실행 세부정보

실행 그래프

| 행 | min_price | max_price | avg_price | |
|---|-----------|-----------|-------------------|--|
| 1 | 0.0 | 649.5 | 2.904956757406... | |

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

```
SELECT COUNT(Quantity) AS cnt_quantity, MIN(Quantity) AS min_quantity, MAX(Quantity) AS max_quantity
FROM hopeful-object-439400-e2.modulabs_project.data
WHERE UnitPrice = 0.0;
```

[결과 이미지를 넣어주세요]

← 쿼리 결과

결과 저장

데이터 탐색

작업 정보

결과

차트

JSON

실행 세부정보

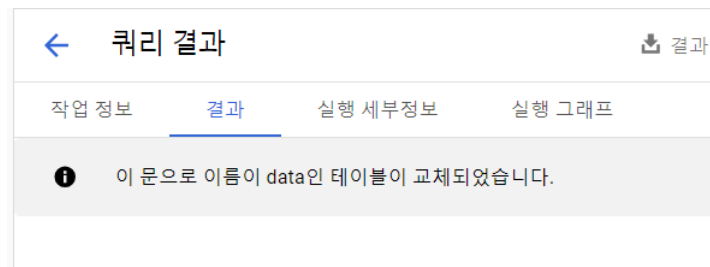
실행 그래프

| 행 | cnt_quantity | min_quantity | max_quantity | avg_quantity |
|---|--------------|--------------|--------------|-------------------|
| 1 | 33 | 1 | 12540 | 420.5151515151... |

- UnitPrice = 0를 제거하고 일관된 데이터셋을 유지하기

```
CREATE OR REPLACE TABLE hopeful-object-439400-e2.modulabs_project.data AS
SELECT *
FROM hopeful-object-439400-e2.modulabs_project.data
WHERE UnitPrice != 0.0;
```

[결과 이미지를 넣어주세요]



11-7. RFM 스코어

Recency

- **InvoiceDate** 컬럼을 연월일 자료형으로 변경하기

```
SELECT DATE(InvoiceDate) AS InvoiceDay, *  
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

A screenshot of a web application interface showing a query result table. The table has 5 columns: '행' (Row), 'InvoiceDay', 'InvoiceNo', 'StockCode', and 'Quantity'. There are 9 rows of data. The 'InvoiceDay' column contains the date '2011-11-03' for all rows. The 'InvoiceNo' column contains '574301' for all rows. The 'StockCode' column contains various codes: '85049A', '85049E', '20749', '84879', '22750', '20971', '23514', '23511', and '22086'. The 'Quantity' column contains values: '12', '12', '4', '8', '4', '12', '6', '6', and '6'. The interface includes a header with '쿼리 결과' (Query Result), a download icon, and a search icon. Below the header is a navigation bar with tabs: '작업 정보' (Job Info), '결과' (Result), '차트' (Chart), 'JSON', '실행 세부정보' (Execution Details), and '실행 그래프' (Execution Graph). The '결과' tab is selected. The table is displayed below the tabs.

- 가장 최근 구매 일자를 MAX() 함수로 찾아보기

```
SELECT  
    MAX(InvoiceDate) OVER () AS most_recent_date,  
    DATE(InvoiceDate) AS InvoiceDay  
FROM hopeful-object-439400-e2.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

쿼리 결과

결과 저장

| 작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프 |
|-------|-------------------------|------------|------|---------|--------|
| 행 | most_recent_date | InvoiceDay | | | |
| 1 | 2011-12-09 12:50:00 UTC | 2011-06-19 | | | |
| 2 | 2011-12-09 12:50:00 UTC | 2011-06-19 | | | |
| 3 | 2011-12-09 12:50:00 UTC | 2011-02-02 | | | |
| 4 | 2011-12-09 12:50:00 UTC | 2011-11-25 | | | |
| 5 | 2011-12-09 12:50:00 UTC | 2010-12-01 | | | |
| 6 | 2011-12-09 12:50:00 UTC | 2011-07-01 | | | |
| 7 | 2011-12-09 12:50:00 UTC | 2011-07-22 | | | |
| 8 | 2011-12-09 12:50:00 UTC | 2011-07-22 | | | |
| 9 | 2011-12-09 12:50:00 UTC | 2011-03-25 | | | |
| 10 | 2011-12-09 12:50:00 UTC | 2011-05-25 | | | |

- 유저 별로 가장 큰 InvoiceDay를 찾아서 가장 최근 구매일로 저장하기

```
SELECT
  CustomerID,
  MAX(DATE(InvoiceDate))AS InvoiceDay
FROM hopeful-object-439400-e2.modulabs_project.data
GROUP BY CustomerID;
```

[결과 이미지를 넣어주세요]

← 쿼리 결과

| 작업 정보 | 결과 | 차트 | JSON |
|-------|------------|------------|------|
| 행 | CustomerID | InvoiceDay | |
| 1 | 12544 | 2011-11-10 | |
| 2 | 13568 | 2011-06-19 | |
| 3 | 13824 | 2011-11-07 | |
| 4 | 14080 | 2011-11-07 | |
| 5 | 14336 | 2011-11-23 | |
| 6 | 14592 | 2011-11-04 | |
| 7 | 15104 | 2011-06-26 | |
| 8 | 15360 | 2011-10-31 | |
| 9 | 15872 | 2011-11-25 | |
| 10 | 16128 | 2011-11-22 | |
| 11 | 16384 | 2011-09-11 | |
| 12 | 17152 | 2011-05-29 | |

- 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 hopeful-object-439400-e2.modulabs_project.data
```

```
GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]

← 쿼리 결과

| 작업 정보 | 결과 | 차트 | JSON |
|-------|------------|---------|------|
| 행 | CustomerID | recency | |
| 1 | 12346 | 325 | |
| 2 | 12347 | 2 | |
| 3 | 12348 | 75 | |
| 4 | 12349 | 18 | |
| 5 | 12350 | 310 | |
| 6 | 12352 | 36 | |
| 7 | 12353 | 204 | |
| 8 | 12354 | 232 | |
| 9 | 12355 | 214 | |
| 10 | 12356 | 22 | |
| 11 | 12357 | 33 | |

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

```
CREATE OR REPLACE TABLE hopeful-object-439400-e2.modulabs_project.user_r AS
SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM hopeful-object-439400-e2.modulabs_project.data
  GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]

| 행 | CustomerID | recency |
|----|------------|---------|
| 1 | 16446 | 0 |
| 2 | 15804 | 0 |
| 3 | 17754 | 0 |
| 4 | 12518 | 0 |
| 5 | 12748 | 0 |
| 6 | 12433 | 0 |
| 7 | 14397 | 0 |
| 8 | 16626 | 0 |
| 9 | 12985 | 0 |
| 10 | 12680 | 0 |
| 11 | 15344 | 0 |
| 12 | 14441 | 0 |
| 13 | 17001 | 0 |
| 14 | 13113 | 0 |
| 15 | 17428 | 0 |
| 16 | 17490 | 0 |

Frequency

- 고객마다 고유한 InvoiceNo의 수를 세어보기

```
SELECT
  CustomerID,
  COUNT (DISTINCT InvoiceNo) AS Purchase_cnt
FROM hopeful-object-439400-e2.modulabs_project.data
GROUP BY CustomerID
```

[결과 이미지를 넣어주세요]

| 쿼리 결과 | | | 결과 저장 |
|---------|------------|--------------|-------|
| 작업 정보 | 결과 | 차트 | JSON |
| 실행 세부정보 | 실행 그래프 | | |
| 행 | CustomerID | Purchase_cnt | |
| 1 | 12544 | 2 | |
| 2 | 13568 | 1 | |
| 3 | 13824 | 5 | |
| 4 | 14080 | 1 | |
| 5 | 14336 | 4 | |
| 6 | 14592 | 3 | |
| 7 | 15104 | 3 | |
| 8 | 15360 | 1 | |
| 9 | 15872 | 2 | |
| 10 | 16128 | 5 | |
| 11 | 16384 | 2 | |
| 12 | 17152 | 4 | |

페이지당 결과 수: 50 ▼ 1 - 50 (전체 4362행)

- 각 고객 별로 구매한 아이템의 총 수량 더하기

```
SELECT
  CustomerID,
  SUM (Quantity) As item_cnt
FROM hopeful-object-439400-e2.modulabs_project.data
GROUP BY CustomerID;
```

[결과 이미지를 넣어주세요]

쿼리 결과

| 작업 정보 | 결과 | 차트 | JSON | 실 |
|-------|------------|----------|------|---|
| 행 | CustomerID | item_cnt | | |
| 9 | 15872 | 187 | | |
| 10 | 16128 | 988 | | |
| 11 | 16384 | 260 | | |
| 12 | 17152 | 477 | | |
| 13 | 17408 | 3 | | |
| 14 | 17664 | 604 | | |
| 15 | 17920 | 2471 | | |
| 16 | 18176 | 279 | | |
| 17 | 12545 | 517 | | |
| 18 | 13313 | 851 | | |
| 19 | 13569 | 155 | | |

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

```
CREATE OR REPLACE TABLE hopeful-object-439400-e2.modulabs_project.user_rf AS

-- (1) 전체 거래 건수 계산
WITH purchase_cnt AS (
  SELECT
    CustomerID,
    COUNT (DISTINCT InvoiceNo) AS Purchase_cnt
  FROM hopeful-object-439400-e2.modulabs_project.data
  GROUP BY CustomerID
),

-- (2) 구매한 아이템 총 수량 계산
item_cnt AS (
  SELECT
    CustomerID,
    SUM (Quantity) As item_cnt
  FROM hopeful-object-439400-e2.modulabs_project.data
  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 hopeful-object-439400-e2.modulabs_project.user_r AS ur
  ON pc.CustomerID = ur.CustomerID;
```

[결과 이미지를 넣어주세요]

user_rf

쿼리

공유

복사

스냅샷

삭제

<

스키마

세부정보

미리보기

테이블 탐색기

미리보기

통계

계보

| 행 | CustomerID | purchase_cnt | item_cnt | recency | |
|----|------------|--------------|----------|---------|--|
| 1 | 12713 | 1 | 505 | 0 | |
| 2 | 15083 | 1 | 38 | 256 | |
| 3 | 18010 | 1 | 60 | 256 | |
| 4 | 12792 | 1 | 215 | 256 | |
| 5 | 13298 | 1 | 96 | 1 | |
| 6 | 13436 | 1 | 76 | 1 | |
| 7 | 15520 | 1 | 314 | 1 | |
| 8 | 14569 | 1 | 79 | 1 | |
| 9 | 13357 | 1 | 321 | 257 | |
| 10 | 14476 | 1 | 110 | 257 | |
| 11 | 14204 | 1 | 72 | 2 | |
| 12 | 15471 | 1 | 256 | 2 | |
| 13 | 15195 | 1 | 1404 | 2 | |
| 14 | 15992 | 1 | 17 | 3 | |
| 15 | 15318 | 1 | 642 | 3 | |
| 16 | 16569 | 1 | 93 | 3 | |

페이지당 결과 수: 50 1 - 50 (전체 4362행)

자어 기

Monetary

- 고객별 총 지출액 계산 (소수점 첫째 자리에서 반올림)

```
SELECT
  CustomerID,
  ROUND(SUM (UnitPrice * Quantity),1) AS user_total
FROM hopeful-object-439400-e2.modulabs_project.data
GROUP BY CustomerID;
```

[결과 이미지를 넣어주세요]

쿼리 결과

작업 정보

결과

차트

JSON

실행 세부정보

| 행 | CustomerID | user_total | |
|---|------------|------------|--|
| 1 | 12544 | 299.7 | |
| 2 | 13568 | 187.1 | |
| 3 | 13824 | 1698.9 | |
| 4 | 14080 | 45.6 | |
| 5 | 14336 | 1614.9 | |
| 6 | 14592 | 557.9 | |
| 7 | 15104 | 968.6 | |
| 8 | 15360 | 427.9 | |
| 9 | 15872 | 316.2 | |

더보기

페이지당 결과 수: 50 1 - 50

- 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 hopeful-object-439400-e2.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 hopeful-object-439400-e2.modulabs_project.user_rf rf
LEFT JOIN (
  -- 고객 별 총 지출액
  SELECT
    CustomerID,
    ROUND(SUM (UnitPrice * Quantity),1) AS user_total
  FROM hopeful-object-439400-e2.modulabs_project.data
  GROUP BY CustomerID
) ut
ON rf.CustomerID = ut.CustomerID;
```

[결과 이미지를 넣어주세요]

←

쿼리 결과

작업 정보

결과

실행 세부정보

실행 그래프

🔔

이 문으로 이름이 user_rfm인 새 테이블이 생성되었습니다.

user_rfm

🔍 쿼리

👤 공유

📄 복사

📷 스냅샷

🗑 삭제

📄 내보내기

스키마

세부정보

미리보기

테이블 탐색기

미리보기

통계

계보

데이터 프로파일

데이터 품질

| 행 | CustomerID | purchase_cnt | item_cnt | recency | user_total | user_average |
|----|------------|--------------|----------|---------|------------|--------------|
| 1 | 12713 | 1 | 505 | 0 | 794.5 | 794.5 |
| 2 | 18010 | 1 | 60 | 256 | 174.8 | 174.8 |
| 3 | 15083 | 1 | 38 | 256 | 88.2 | 88.2 |
| 4 | 12792 | 1 | 215 | 256 | 344.5 | 344.5 |
| 5 | 15520 | 1 | 314 | 1 | 343.5 | 343.5 |
| 6 | 13298 | 1 | 96 | 1 | 360.0 | 360.0 |
| 7 | 14569 | 1 | 79 | 1 | 227.4 | 227.4 |
| 8 | 13436 | 1 | 76 | 1 | 196.9 | 196.9 |
| 9 | 13357 | 1 | 321 | 257 | 609.4 | 609.4 |
| 10 | 14476 | 1 | 110 | 257 | 193.0 | 193.0 |
| 11 | 15195 | 1 | 1404 | 2 | 3861.0 | 3861.0 |
| 12 | 14204 | 1 | 72 | 2 | 150.6 | 150.6 |
| 13 | 15471 | 1 | 256 | 2 | 454.5 | 454.5 |
| 14 | 16569 | 1 | 93 | 3 | 124.2 | 124.2 |
| 15 | 17914 | 1 | 457 | 3 | 329.4 | 329.4 |
| 16 | 12650 | 1 | 250 | 3 | 242.4 | 242.4 |
| 17 | 15318 | 1 | 642 | 3 | 312.6 | 312.6 |
| 18 | 16528 | 1 | 171 | 3 | 244.4 | 244.4 |

RFM 통합 테이블 출력하기

- 최종 `user_rfm` 테이블을 출력하기


```
SELECT *
FROM hopeful-object-439400-e2.modulabs_project.user_rfm
```

[결과 이미지를 넣어주세요]

← 쿼리 결과 결과 저장 대

작업 정보 **결과** **자트** **JSON** **실행 세부정보** **실행 그래프**

| 행 | CustomerID | purchase_cnt | item_cnt | recency | user_total | user_average |
|----|------------|--------------|----------|---------|------------|--------------|
| 1 | 12713 | 1 | 505 | 0 | 794.5 | 794.5 |
| 2 | 18010 | 1 | 60 | 256 | 174.8 | 174.8 |
| 3 | 12792 | 1 | 215 | 256 | 344.5 | 344.5 |
| 4 | 15082 | 1 | 38 | 256 | 88.2 | 88.2 |
| 5 | 15520 | 1 | 314 | 1 | 343.5 | 343.5 |
| 6 | 14569 | 1 | 79 | 1 | 227.4 | 227.4 |
| 7 | 13436 | 1 | 76 | 1 | 196.9 | 196.9 |
| 8 | 13298 | 1 | 96 | 1 | 360.0 | 360.0 |
| 9 | 13357 | 1 | 321 | 257 | 609.4 | 609.4 |
| 10 | 14476 | 1 | 110 | 257 | 193.0 | 193.0 |
| 11 | 15471 | 1 | 256 | 2 | 454.5 | 454.5 |
| 12 | 14204 | 1 | 72 | 2 | 150.6 | 150.6 |

페이지당 결과 수: 50 1 ~ 50 (합계 439400) 1K

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]

user_data 쿼리 공유 복사 스냅샷 삭제 내보내기

스키마 세부정보 **미리보기** **테이블 탐색기** **테이블 보기** **종계** **계보** **데이터 프로파일** **데이터 용**

| 행 | CustomerID | purchase_cnt | item_cnt | recency | user_total | user_average | unique_products |
|----|------------|--------------|----------|---------|------------|--------------|-----------------|
| 1 | 17986 | 1 | 10 | 56 | 20.8 | 20.8 | 1 |
| 2 | 17752 | 1 | 192 | 359 | 80.6 | 80.6 | 1 |
| 3 | 13017 | 1 | 48 | 7 | 204.0 | 204.0 | 1 |
| 4 | 16344 | 1 | 18 | 158 | 101.1 | 101.1 | 1 |
| 5 | 13120 | 1 | 12 | 238 | 30.6 | 30.6 | 1 |
| 6 | 12814 | 1 | 48 | 101 | 85.9 | 85.9 | 1 |
| 7 | 13841 | 1 | 100 | 252 | 85.0 | 85.0 | 1 |
| 8 | 13829 | 1 | -12 | 359 | -102.0 | -102.0 | 1 |
| 9 | 13270 | 1 | 200 | 366 | 590.0 | 590.0 | 1 |
| 10 | 13307 | 1 | 4 | 120 | 15.0 | 15.0 | 1 |
| 11 | 14351 | 1 | 12 | 164 | 51.0 | 51.0 | 1 |
| 12 | 14679 | 1 | -1 | 371 | -2.5 | -2.5 | 1 |
| 13 | 17948 | 1 | 144 | 147 | 358.6 | 358.6 | 1 |
| 14 | 15657 | 1 | 24 | 22 | 30.0 | 30.0 | 1 |
| 15 | 17102 | 1 | 2 | 261 | 25.5 | 25.5 | 1 |

쿼리 결과

결과

작업 정보 **결과** 실행 세부정보 실행 그래프

이 문으로 이름이 user_data인 새 테이블이 생성되었습니다.

2. 평균 구매 주기

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

```
CREATE OR REPLACE TABLE hopeful-object-439400-e2.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_inte
  FROM (
    -- (1) 구매와 구매 사이에 소요된 일수
    SELECT
      CustomerID,
      DATE_DIFF(InvoiceDate, LAG(InvoiceDate) OVER (PARTITION BY CustomerID ORDER BY InvoiceDate), DAY)
    FROM
      hopeful-object-439400-e2.modulabs_project.data
    WHERE CustomerID IS NOT NULL
  )
  GROUP BY CustomerID
)

SELECT u.*, pi.* EXCEPT (CustomerID)
FROM hopeful-object-439400-e2.modulabs_project.user_data AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;
```

[결과 이미지를 넣어주세요]

| 스키마 | 세부정보 | 타리보기 | 데이터를 탐색하기 | 필터링 | 통계 | 정보 | 데이터의 흐름 | 데이터 중심 |
|-----|------------|--------------|-----------|---------|------------|--------------|----------------|------------------|
| id | CustomerID | purchase_cnt | item_cnt | recency | user_total | user_average | unique_product | average_interval |
| 1 | 14432 | 6 | 2013 | 9 | 2248.5 | 374.75 | 256 | 0.2 |
| 2 | 12428 | 11 | 3477 | 25 | 6366.0 | 578.727272 | 256 | 0.87 |
| 3 | 13268 | 14 | 3525 | 17 | 3105.7 | 221.833714 | 256 | 0.56 |
| 4 | 16454 | 1 | 2 | 64 | 5.9 | 5.9 | 1 | 0.0 |
| 5 | 15753 | 1 | 144 | 304 | 79.2 | 79.2 | 1 | 0.0 |
| 6 | 13099 | 1 | 288 | 99 | 207.4 | 207.4 | 1 | 0.0 |
| 7 | 15195 | 1 | 1404 | 2 | 3861.0 | 3861.0 | 1 | 0.0 |
| 8 | 17715 | 1 | 384 | 200 | 326.4 | 326.4 | 1 | 0.0 |
| 9 | 16138 | 1 | -1 | 368 | -8.0 | -8.0 | 1 | 0.0 |
| 10 | 16344 | 1 | 18 | 198 | 101.1 | 101.1 | 1 | 0.0 |
| 11 | 15668 | 1 | 72 | 217 | 76.3 | 76.3 | 1 | 0.0 |
| 12 | 13366 | 1 | 144 | 50 | 56.2 | 56.2 | 1 | 0.0 |
| 13 | 14679 | 1 | -1 | 371 | -2.5 | -2.5 | 1 | 0.0 |
| 14 | 18293 | 1 | 4 | 325 | 440.0 | 440.0 | 1 | 0.0 |
| 15 | 16323 | 1 | 50 | 196 | 207.5 | 207.5 | 1 | 0.0 |
| 16 | 18141 | 1 | -12 | 360 | -35.4 | -35.4 | 1 | 0.0 |
| 17 | 15498 | 1 | 72 | 92 | 76.3 | 76.3 | 1 | 0.0 |
| 18 | 18184 | 1 | 60 | 15 | 49.8 | 49.8 | 1 | 0.0 |

| 쿼리 결과 | 결과 저장 |
|------------------------------------|-------|
| 작업 정보 | 결과 |
| 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 hopeful-object-439400-e2.modulabs_project.user_data AS

WITH TransactionInfo AS (
  SELECT
    CustomerID,
    COUNT(InvoiceNo) AS total_transactions,
    COUNT(CASE WHEN InvoiceNo LIKE 'C%' THEN 1 ELSE 0 END) AS cancel_frequency
  FROM hopeful-object-439400-e2.modulabs_project.data
  GROUP BY CustomerID
)

SELECT u.*, t.* EXCEPT(CustomerID), ROUND((cancel_frequency / total_transactions) * 100 , 2) AS cancel_
```

```
FROM hopeful-object-439400-e2.modulabs_project.data AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID;
```

[결과 이미지를 넣어주세요]

| id | InvoiceNo | StockCode | Quantity | InvoiceDate | UnitPrice | CustomerID | Country | Description | total | tax | cancel |
|----|-----------|-----------|----------|-------------------------|-----------|------------|----------------|-------------------------------|-------|-----|--------|
| 11 | 574301 | 22734 | 6 | 2011-11-03 16:15:00 UTC | 2.89 | 12544 | Spain | SET OF 4 RIBBONS VINTAGE C... | 19 | 1 | 5.26 |
| 12 | 574301 | 20749 | 4 | 2011-11-03 16:15:00 UTC | 7.95 | 12544 | Spain | ASSORTED COLOUR MINI CAS... | 19 | 1 | 5.26 |
| 13 | 574301 | 23314 | 6 | 2011-11-03 16:15:00 UTC | 2.08 | 12544 | Spain | EMBROIDERED RIBBON REEL S... | 19 | 1 | 5.26 |
| 14 | 574301 | 94879 | 8 | 2011-11-03 16:15:00 UTC | 1.99 | 12544 | Spain | ASSORTED COLOUR BIRD ORN... | 19 | 1 | 5.26 |
| 15 | 574301 | 22086 | 6 | 2011-11-03 16:15:00 UTC | 2.95 | 12544 | Spain | PAPER CHAIN KIT S/S CHRIST... | 19 | 1 | 5.26 |
| 16 | 574301 | 22750 | 4 | 2011-11-03 16:15:00 UTC | 3.75 | 12544 | Spain | FELTDRAFT PRINCESS LOLA O... | 19 | 1 | 5.26 |
| 17 | 574301 | 22144 | 6 | 2011-11-03 16:15:00 UTC | 2.1 | 12544 | Spain | CHRISTMAS CRAFT LITTLE FRL... | 19 | 1 | 5.26 |
| 18 | 574301 | 22077 | 12 | 2011-11-03 16:15:00 UTC | 1.95 | 12544 | Spain | 6 RIBBONS RUSTIC CHARM | 19 | 1 | 5.26 |
| 19 | 0357531 | 22960 | 4 | 2011-11-10 11:12:00 UTC | 4.25 | 12544 | Spain | JAM MAKING SET WITH JARS | 19 | 1 | 5.26 |
| 20 | 557305 | 23302 | 1 | 2011-06-19 14:42:00 UTC | 1.65 | 13568 | United Kingdom | KNEELING MAT HOUSEWORK ... | 43 | 0 | 0.0 |
| 21 | 557305 | 23268 | 2 | 2011-06-19 14:42:00 UTC | 4.95 | 13568 | United Kingdom | SPOTTY BUNTING | 43 | 0 | 0.0 |
| 22 | 557305 | 22668 | 1 | 2011-06-19 14:42:00 UTC | 2.95 | 13568 | United Kingdom | PINK BABY BUNTING | 43 | 0 | 0.0 |
| 23 | 557305 | 22460 | 2 | 2011-06-19 14:42:00 UTC | 1.65 | 13568 | United Kingdom | HEART OF RICKER SMALL | 43 | 0 | 0.0 |
| 24 | 557305 | 171070 | 2 | 2011-06-19 14:42:00 UTC | 2.55 | 13568 | United Kingdom | FLOWER FAIRY'S SUMMER BO... | 43 | 0 | 0.0 |
| 25 | 557305 | 21429 | 1 | 2011-06-19 14:42:00 UTC | 1.95 | 13568 | United Kingdom | RED BINGHAM ROSE JEWELLE... | 43 | 0 | 0.0 |
| 26 | 557305 | 23254 | 1 | 2011-06-19 14:42:00 UTC | 4.15 | 13568 | United Kingdom | CHILDRENS OUTLRY DOLLY G... | 43 | 0 | 0.0 |
| 27 | 557305 | 23268 | 1 | 2011-06-19 14:42:00 UTC | 4.95 | 13568 | United Kingdom | SPOTTY BUNTING | 43 | 0 | 0.0 |
| 28 | 557305 | 23561 | 1 | 2011-06-19 14:42:00 UTC | 1.65 | 13568 | United Kingdom | WOODEN SCHOOL COLOURIN... | 43 | 0 | 0.0 |
| 29 | 557305 | 23167 | 1 | 2011-06-19 14:42:00 UTC | 9.95 | 13568 | United Kingdom | OVAL WALL MIRROR DIAMAN... | 43 | 0 | 0.0 |
| 30 | 557305 | 21733 | 4 | 2011-06-19 14:42:00 UTC | 2.95 | 13568 | United Kingdom | RED HANGING HEART FLIGHT ... | 43 | 0 | 0.0 |

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

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

[결과 이미지를 넣어주세요]

회고

마지막 문제를 여러방면으로 시도해 보았는데 결과가 원하는 대로 표가 합쳐지지 않은 듯 해서 다시 한 번 시도해 보려고 합니다.

- FROM뒤에 user_data 와 본래 주소 가 혼돈되었던 듯 합니다.