11. 고객 세그멘테이션 프로젝트

```
-- -- 컬럼 별 누락된 값의 비율 계산
SELECT column_name, ROUND((total - column_value) / total * 100, 2)
FROM
(
    SELECT 'InvoiceNo' AS column_name, COUNT(InvoiceNo) AS column_value
    SELECT 'StockCode' AS column_name, COUNT(StockCode) AS column_val
    SELECT 'Description' AS column_name, COUNT(Description) AS column_va
    SELECT 'Quantity' AS column_name, COUNT(Quantity) AS column_value, C
    SELECT 'InvoiceDate' AS column_name, COUNT(InvoiceDate) AS column_v
    SELECT 'UnitPrice' AS column_name, COUNT(UnitPrice) AS column_value,
    SELECT 'CustomerID' AS column_name, COUNT(CustomerID) AS column_v
    SELECT 'Country' AS column_name, COUNT(Country) AS column_value, CO
) AS column_data;
```

행	column_name	f0_
1	InvoiceNo	0.0
2	UnitPrice	0.0
3	Description	0.27
4	StockCode	0.0
5	CustomerID	24.93
6	Country	0.0
7	InvoiceDate	0.0
8	Quantity	0.0

```
-- 결측치 있는 행 제거
DELETE FROM hardy-aleph-464902-v8.modulabs_project.data4
WHERE Description IS NULL
  OR CustomerID IS NULL
```

이 문으로 data4의 행 135,080개가 삭제되었습니다.

```
-- 중복된 행의 개수 확인
SELECT InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice,
```

```
FROM hardy-aleph-464902-v8.modulabs_project.data4
GROUP BY InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPric
HAVING COUNT(*) > 1
```

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	571034	23494	VINTAGE DOILY DELUXE SEWIN...	3	2011-10-13 12:47:00 UTC	5.95	12359	Cyprus
2	571034	23245	SET OF 3 REGENCY CAKE TINS	4	2011-10-13 12:47:00 UTC	4.95	12359	Cyprus
3	571034	23239	SET OF 4 KNICK KNACK TINS P...	6	2011-10-13 12:47:00 UTC	4.15	12359	Cyprus
4	538826	22749	FELTCRAFT PRINCESS CHARLO...	1	2010-12-14 12:58:00 UTC	3.75	12370	Cyprus
5	577228	22435	SET OF 9 HEART SHAPED BALL...	1	2011-11-18 12:07:00 UTC	1.25	12391	Cyprus
6	577228	84580	MOUSE TOY WITH PINK T-SHIRT	1	2011-11-18 12:07:00 UTC	3.75	12391	Cyprus
7	577228	22270	HAPPY EASTER HANGING DEC...	1	2011-11-18 12:07:00 UTC	3.75	12391	Cyprus
8	577228	23156	SET OF 5 MINI GROCERY MAG...	1	2011-11-18 12:07:00 UTC	2.08	12391	Cyprus

이 문으로 이름이 data4인 테이블이 교체되었습니다.

```
-- 중복값을 처리하고 난 후 남은 데이터의 행의 개수
SELECT COUNT(*)
FROM hardy-aleph-464902-v8.modulabs_project.data4
```

행	f0_
1	401604

```
-- -- 고유(unique)한 InvoiceNo의 개수를 출력
SELECT DISTINCT InvoiceNo
FROM hardy-aleph-464902-v8.modulabs_project.data4
LIMIT 100
```

행	InvoiceNo
5	549222
6	556201
7	562032
8	573511
9	581180
10	539318
11	541998
12	548955

```
-- InvoiceNo가 'C'로 시작하는 행을 필터링
SELECT DISTINCT InvoiceNo, *
FROM hardy-aleph-464902-v8.modulabs_project.data4
WHERE InvoiceNo LIKE 'C%'
LIMIT 100
```

행	InvoiceNo	InvoiceNo_1	StockCode	Description	Quantity	InvoiceDate	UnitPrice
5	C547388	C547388	22784	LANTERN CREAM GAZEBO	-3	2011-03-22 16:07:00 UTC	4
6	C547388	C547388	37448	CERAMIC CAKE DESIGN SPOTT...	-12	2011-03-22 16:07:00 UTC	1
7	C547388	C547388	22701	PINK DOG BOWL	-6	2011-03-22 16:07:00 UTC	2
8	C547388	C547388	22645	CERAMIC HEART FAIRY CAKE ...	-12	2011-03-22 16:07:00 UTC	1
9	C547388	C547388	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2
10	C547388	C547388	84050	PINK HEART SHAPE EGG FRYIN...	-12	2011-03-22 16:07:00 UTC	1
11	C547388	C547388	21914	BLUE HARMONICA IN BOX	-12	2011-03-22 16:07:00 UTC	1
12	C549955	C549955	22839	3 TIER CAKE TIN GREEN AND C...	-2	2011-04-13 13:38:00 UTC	14

```
-- 구매 건 상태가 Canceled 인 데이터의 비율
SELECT ROUND(SUM(CASE WHEN InvoiceNo LIKE 'C%' THEN 1 ELSE 0 END)
FROM hardy-aleph-464902-v8.modulabs_project.data4
```

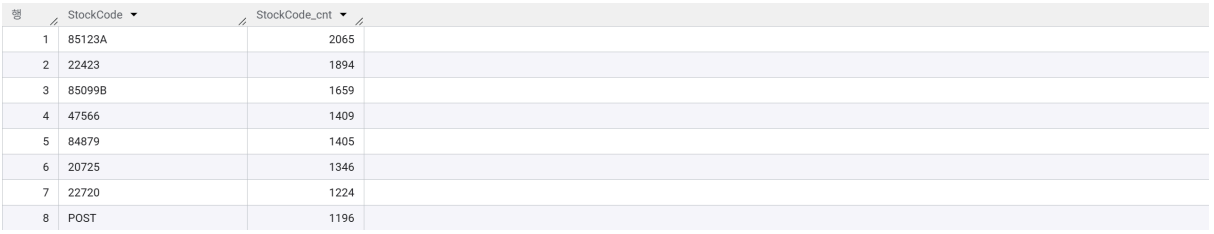
행	f0_
1	2.2

```
-- 고유한 StockCode의 개수를 출력
SELECT COUNT(DISTINCT StockCode)
FROM hardy-aleph-464902-v8.modulabs_project.data1
```

행	f0_
1	3684

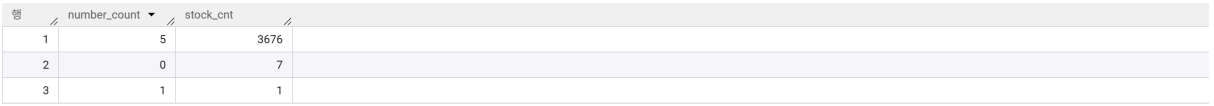
```
-- StockCode 별 등장 빈도를 출력
SELECT StockCode, COUNT(StockCode) AS StockCode_cnt
FROM hardy-aleph-464902-v8.modulabs_project.data4
```

```
GROUP BY StockCode
ORDER BY StockCode_cnt DESC
LIMIT 10
```



행	StockCode	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

```
-- StockCode의 문자열 내 숫자의 길이
WITH UniqueStockCodes AS (
  SELECT DISTINCT StockCode
  FROM hardy-aleph-464902-v8.modulabs_project.data4
)
SELECT
  LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) A
  COUNT(*) AS stock_cnt
FROM UniqueStockCodes
GROUP BY number_count
ORDER BY stock_cnt DESC;
```



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

```
-- 숫자가 0~1개인 값들에는 어떤 코드들이 들어가 있는지를 확인
SELECT DISTINCT StockCode, number_count
FROM (
  SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', ''))
  FROM hardy-aleph-464902-v8.modulabs_project.data4
)
WHERE number_count = 0 OR number_count = 1
```

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

```
-- 해당 코드 값들을 가지고 있는 데이터 수는 전체 데이터 수 대비 몇 퍼센트?
SELECT ROUND(SUM(CASE WHEN StockCode IN ('POST','D','C2', 'M', 'BANK
FROM hardy-aleph-464902-v8.modulabs_project.data4
```

행	f0_
1	0.48

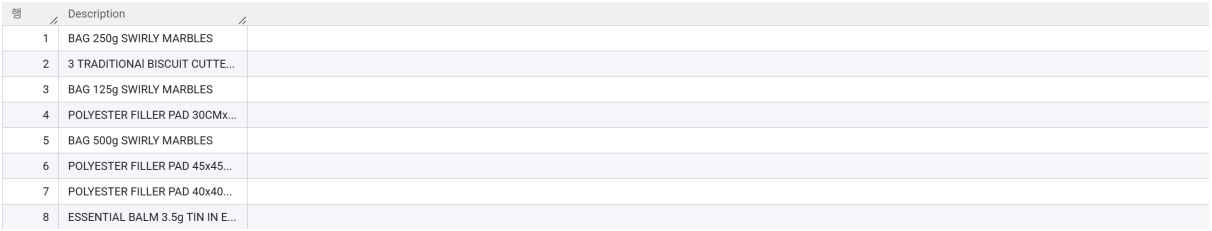
```
-- 제품과 관련되지 않은 거래 기록을 제거하는 쿼리문
DELETE FROM hardy-aleph-464902-v8.modulabs_project.data4
WHERE StockCode IN ('POST','D','C2', 'M', 'BANK CHARGES', 'PADS', 'DOT', '(
```

이 문으로 data4의 행 1,915개가 삭제되었습니다.

```
-- 고유한 Description 별 출현 빈도를 계산하고 상위 30개를 출력
SELECT Description, COUNT(*) AS description_cnt
FROM hardy-aleph-464902-v8.modulabs_project.data4
GROUP BY Description
LIMIT 30;
```

행	Description	description_cnt
1	MEDIUM CERAMIC TOP STORA...	208
2	RED TOADSTOOL LED NIGHT LI...	539
3	AIRLINE BAG VINTAGE JET SET...	96
4	ALARM CLOCK BAKELIKE RED	917
5	CAMOUFLAGE EAR MUFF HEA...	17
6	CLEAR DRAWER KNOB ACRYLI...	331
7	BLUE DRAWER KNOB ACRYLIC ...	124
8	BLACK GRAND BAROQUE PHOT...	7

```
-- 대소문자가 혼합된 Description이 있는지 확인
SELECT DISTINCT Description
FROM hardy-aleph-464902-v8.modulabs_project.data4
WHERE REGEXP_CONTAINS(Description, r'[a-z]')
```



행	Description
1	BAG 250g SWIRLY MARBLES
2	3 TRADITIONAI BISCUIT CUTTE...
3	BAG 125g SWIRLY MARBLES
4	POLYESTER FILLER PAD 30CMx...
5	BAG 500g SWIRLY MARBLES
6	POLYESTER FILLER PAD 45x45...
7	POLYESTER FILLER PAD 40x40...
8	ESSENTIAL BALM 3.5g TIN IN E...

```
-- -- 서비스 관련 정보를 포함하는 행들을 제거
DELETE
FROM hardy-aleph-464902-v8.modulabs_project.data4
WHERE Description IN ('High Resolution Image','Next Day Carriage')
```

이 문으로 data4의 행 83개가 삭제되었습니다.

```
-- 대소문자를 혼합하고 있는 데이터를 대문자로 표준화
CREATE OR REPLACE TABLE hardy-aleph-464902-v8.modulabs_project.data4
SELECT
  * EXCEPT (Description),
  UPPER(Description) AS Description
FROM hardy-aleph-464902-v8.modulabs_project.data4
```

이 문으로 이름이 data4인 테이블이 교체되었습니다.

```
-- UnitPrice의 최솟값, 최댓값, 평균
SELECT MIN(UnitPrice) AS min_price, MAX(UnitPrice) AS max_price, AVG(Uni
```

```
FROM hardy-aleph-464902-v8.modulabs_project.d
```

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
SELECT COUNT(Quantity) AS cnt_quantity, MIN(Quantity) AS min_quantity, MA
FROM hardy-aleph-464902-v8.modulabs_project.data4
WHERE UnitPrice = 0
```

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

```
-- --  DATE 함수를 활용하여 InvoiceDate 컬럼을 연월일 자료형으로 변경
SELECT DATE(InvoiceDate) AS InvoiceDay, *
FROM hardy-aleph-464902-v8.modulabs_project.data4
ORDER BY InvoiceDay DESC
```

행	InvoiceDay	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	2011-12-09	581493	79190B	15	2011-12-09 10:10:00 UTC	0.42	12423	Belgium
2	2011-12-09	581493	22807	6	2011-12-09 10:10:00 UTC	2.95	12423	Belgium
3	2011-12-09	581493	22252	12	2011-12-09 10:10:00 UTC	1.25	12423	Belgium
4	2011-12-09	581493	22915	12	2011-12-09 10:10:00 UTC	0.42	12423	Belgium
5	2011-12-09	581493	22356	10	2011-12-09 10:10:00 UTC	0.85	12423	Belgium
6	2011-12-09	581493	22865	12	2011-12-09 10:10:00 UTC	2.1	12423	Belgium
7	2011-12-09	581493	84945	12	2011-12-09 10:10:00 UTC	0.85	12423	Belgium
8	2011-12-09	581493	22632	12	2011-12-09 10:10:00 UTC	2.1	12423	Belgium

```
-- 가장 최근 구매 일자 MAX() 함수로 찾기
SELECT MAX(DATE(InvoiceDate)) AS most_recent_date
FROM `hardy-aleph-464902-v8.modulabs_project.data4`;
```

행	most_recent_date
1	2011-12-09

```
-- 유저 별로 가장 큰 InvoiceDay를 찾아서 가장 최근 구매일로 저장
SELECT
  CustomerID,
  DATE(MAX(InvoiceDate)) AS most_recent_date
FROM hardy-aleph-464902-v8.modulabs_project.data4
GROUP BY CustomerID
```

행	CustomerID	most_recent_date
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

```
-- 가장 최근 일자(most_recent_date)와 유저별 마지막 구매일(InvoiceDay)간의 차이를
SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM hardy-aleph-464902-v8.modulabs_project.data4
  GROUP BY CustomerID
);
```

행	CustomerID	recency
1	12438	14
2	12534	130
3	12600	28
4	12723	3
5	12812	44

```
-- 전체 거래 건수 계산
SELECT
  CustomerID,
  COUNT(DISTINCT InvoiceNo) AS purchase_cnt
FROM hardy-aleph-464902-v8.modulabs_project.data4
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

```
-- 구매한 아이템의 총 수량 계산
SELECT
  CustomerID,
  SUM(Quantity) AS item_cnt
FROM hardy-aleph-464902-v8.modulabs_project.data4
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

```
CREATE OR REPLACE TABLE hardy-aleph-464902-v8.modulabs_project.user_

-- (1) 전체 거래 건수 계산
WITH purchase_cnt AS (
  SELECT
    CustomerID,
  COUNT(DISTINCT InvoiceNo) AS purchase_cnt
  FROM hardy-aleph-464902-v8.modulabs_project.data4
```

```
  GROUP BY CustomerID
),

-- (2) 구매한 아이템 총 수량 계산
item_cnt AS (
  SELECT
    CustomerID,
  SUM(Quantity) AS item_cnt
  FROM hardy-aleph-464902-v8.modulabs_project.data4
  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 hardy-aleph-464902-v8.modulabs_project.user_r AS ur
  ON pc.CustomerID = ur.CustomerID;
```

이 문으로 이름이 user_rf3인 새 테이블이 생성되었습니다.

```
-- 고객별 총 지출액 계산
SELECT
  CustomerID,
  SUM(Quantity * UnitPrice) AS user_total
FROM hardy-aleph-464902-v8.modulabs_project.data4
GROUP BY CustomerID
```

행	CustomerID	user_total
1	12346	0.0
2	12347	4309.999999999...
3	12348	1437.239999999...
4	12349	1457.549999999...
5	12350	294.4
6	12352	1265.410000000...
7	12353	89.0
8	12354	1079.4

```
-- -- 고객별 평균 거래 금액 계산
CREATE OR REPLACE TABLE `hardy-aleph-464902-v8.modulabs_project.user
SELECT
  rf.CustomerID AS CustomerID,
  rf.purchase_cnt,
  rf.item_cnt,
  rf.recency,
  ut.user_total,
  SAFE_DIVIDE(ut.user_total, rf.purchase_cnt) AS user_average
FROM `hardy-aleph-464902-v8.modulabs_project.user_rf3` AS rf
LEFT JOIN (
  -- 고객 별 총 지출액
  SELECT
    CustomerID,
    SUM(Quantity * UnitPrice) AS user_total
  FROM `hardy-aleph-464902-v8.modulabs_project.data4`
  GROUP BY CustomerID
) ut
ON rf.CustomerID = ut.CustomerID;
```

이 문으로 이름이 user_rfm2인 새 테이블이 생성되었습니다.

```
-- 구매하는 제품의 다양성
CREATE OR REPLACE TABLE hardy-aleph-464902-v8.modulabs_project.user_
WITH unique_products AS (
  SELECT
    CustomerID,
```

```
    COUNT(DISTINCT StockCode) AS unique_products
  FROM hardy-aleph-464902-v8.modulabs_project.data4
  GROUP BY CustomerID
)
SELECT ur.*, up.* EXCEPT (CustomerID)
FROM hardy-aleph-464902-v8.modulabs_project.user_rfm2 AS ur
JOIN unique_products AS up
ON ur.CustomerID = up.CustomerID;
```

이 문으로 이름이 user_data2인 새 테이블이 생성되었습니다.

```
-- 평균 구매 주기
CREATE OR REPLACE TABLE hardy-aleph-464902-v8.modulabs_project.user_
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
  FROM (
    -- (1) 구매와 구매 사이에 소요된 일수
    SELECT
      CustomerID,
      DATE_DIFF(InvoiceDate, LAG(InvoiceDate) OVER (PARTITION BY Custome
    FROM
      hardy-aleph-464902-v8.modulabs_project.data4
    WHERE CustomerID IS NOT NULL
  )
  GROUP BY CustomerID
)

SELECT u.*, pi.* EXCEPT (CustomerID)
FROM hardy-aleph-464902-v8.modulabs_project.user_data2 AS u
```

```
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;
```

이 문으로 이름이 user_data2인 테이블이 교체되었습니다.

```
-- 취소 비율
CREATE OR REPLACE TABLE hardy-aleph-464902-v8.modulabs_project.user_

WITH TransactionInfo AS (
  SELECT
    CustomerID,
    COUNT(DISTINCT InvoiceNo) AS total_transactions,
    COUNT(DISTINCT IF(STARTS_WITH(InvoiceNo, 'C'), InvoiceNo, NULL)) AS
  FROM hardy-aleph-464902-v8.modulabs_project.data4
   WHERE CustomerID IS NOT NULL
  GROUP BY CustomerID
)

SELECT u.*, t.* EXCEPT(CustomerID), SAFE_DIVIDE(t.cancel_frequency, t.tota
FROM `hardy-aleph-464902-v8.modulabs_project.user_data2` AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID;
```

이 문으로 이름이 user_data2인 테이블이 교체되었습니다.