

# Main Quest 2 템플릿

# 평가 항목

채점 기준	설명
1. 추천 시스템의 창의성과 복잡성	5가지의 추천 시스템 아이디어가 모두 독창적인가?
2. SQL 구현의 정확성 및 효율성	SQL 쿼리가 데이터의 목적에 맞게 설계되었는가?
3. 보고서 구성 및 설명력	논리적으로 추천 시스템 구현 로직을 설명했는가?

# 추천시스템 1

### 1. 추천시스템의 이름

• 할인율이 높은 인기 제품 추천

#### 2. 추천시스템의 테마 및 목적

• 아마존 홈페이지 배너에 활용가능한 높은 할인율을 가진 인기 제품 추천하여 고객 유도하기

### 3. 구현 로직 (코드, 쿼리 및 구조도)

•

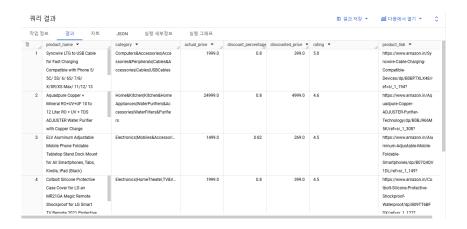
```
SELECT DISTINCT product_name,
    category,
    actual_price,
    discount_percentage,
    discounted_price,
    rating,product_link

FROM `modulabs.amazon2`,
    UNNEST(SPLIT(category, '|')) AS category_value
    WHERE discount_percentage >= 0.8 AND CAST(rating AS FLOAT64) >= 4.5

ORDER BY CAST(rating AS FLOAT64) DESC ,discount_percentage DESC
```

#### 4. 결과

.



### 5. 참고자료

•

# 추천시스템 2

#### 1. 추천시스템의 이름

• 베스트 리뷰어 챌린지

### 2. 추천시스템의 테마 및 목적

• "삼성제품을 구매한 고객 중 50자이상의 리뷰를 작성한 사람 중 베스트 리뷰어 찾기" 고객의 적극적인 참여를 유도해 사용자 생성 콘텐츠를 통해 특정 브랜드 제품 추천하기

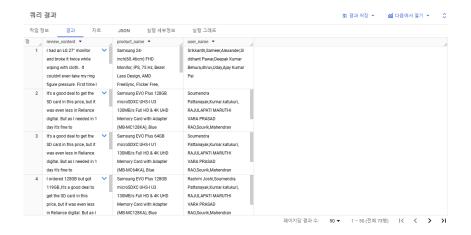
### 3. 구현 로직 (코드, 쿼리 및 구조도)

•

```
SELECT DISTINCT review_content, product_name, user_name
FROM `modulabs.amazon2`
WHERE product_name LIKE "%Samsung%"
AND LENGTH(review_content) >= 50
```

### 4. 결과

•



#### 5. 참고자료

•

# 추천시스템 3

# 1. 추천시스템의 이름

• 고객별 가장 많이 소비한 브랜드 중 평점 4.5이상의 제품들 추천하기

# 2. 추천시스템의 테마 및 목적

• 고객별 가장 많이 소비한 브랜드 중 평점 4.5이상의 제품들 추천하기

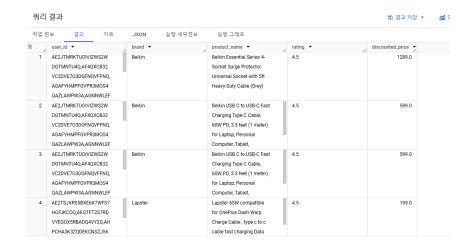
# 3. 구현 로직 (코드, 쿼리 및 구조도)

•

```
WITH customer_top_brand AS (
  SELECT
   user_id,
   SPLIT(product_name, ' ')[0FFSET(0)] AS brand,
   COUNT(*) as purchase_count,
   ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY COUNT(*) DESC) as brand_rank
  FROM `modulabs.amazon2`
 GROUP BY user_id, SPLIT(product_name, ' ')[0FFSET(0)]
)
SELECT
 ctb.user_id,
 ctb.brand,
 a.product_name,
 a.rating,
 a.discounted_price
FROM customer_top_brand ctb
JOIN `modulabs.amazon2` a ON SPLIT(a.product_name, ' ')[OFFSET(0)] = ctb.brand
WHERE ctb.brand_rank = 1
 AND a.rating IS NOT NULL
 AND SAFE_CAST(a.rating AS FLOAT64) IS NOT NULL
 AND SAFE_CAST(a.rating AS FLOAT64) >= 4.5
ORDER BY ctb.user_id, SAFE_CAST(a.rating AS FLOAT64) DESC
```

#### 4. 결과

•



# 5. 참고자료

.

# 추천시스템 4

# 1. 추천시스템의 이름

Main Quest 2 템플릿 3

• 카테고리 별로 할인율 높은 순으로 보여주기

#### 2. 추천시스템의 테마 및 목적

고객이 필요한 카테고리에서 할인률 높은순으로 저렴하게 구매하기 위한 추천

# 3. 구현 로직 (코드, 쿼리 및 구조도)

•

SELECT category, product\_id, product\_name, discounted\_price, discount\_percentage FROM `modulabs.amazon2`
GROUP BY category, product\_id, product\_name, discounted\_price, discount\_percentage

GROUP BY category, product\_id, product\_name, discounted\_price, discount\_percentage ORDER BY category, discount\_percentage DESC

### 4. 결과

•



# 5. 참고자료

•

# 추천시스템 5

# 1. 추천시스템의 이름

• 고객이 자주 구매하는 category의 제품을 우선적으로 추천

#### 2. 추천시스템의 테마 및 목적

•

# 3. 구현 로직 (코드, 쿼리 및 구조도)

•

```
WITH customer_category_purchases AS (
    SELECT
    user_id,
```

Main Quest 2 템플릿 4

```
category,
  COUNT(*) as purchase_count,
  ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY COUNT(*) DESC) as category_rank
FROM `modulabs.amazon2`
  GROUP BY user_id, category
)
SELECT
  user_id,
  category,
  purchase_count
FROM customer_category_purchases
WHERE category_rank = 1
ORDER BY purchase_count DESC, user_id
```

#### 4. 결과

.

### 쿼리 결과



#### 5. 참고자료

.

# 회고

조금 더 다양하게 생각해보고 싶은데 SQL을 잘몰라서 다양해하게 할 수록 오류가 많이 났습니다

Main Quest 2 템플릿 5