

Main Quest 2 템플릿

평가 항목

추천시스템 1

- 1. 추천시스템의 이름
- 2. 추천시스템의 테마 및 목적
- 3. 구현 로직 (코드, 쿼리 및 구조도)
- 4. 결과
- 5. 참고자료

추천시스템 2

- 1. 추천시스템의 이름
- 2. 추천시스템의 테마 및 목적
- 3. 구현 로직 (코드, 쿼리 및 구조도)
- 4. 결과
- 5. 참고자료

추천시스템 3

- 1. 추천시스템의 이름
- 2. 추천시스템의 테마 및 목적
- 3. 구현 로직 (코드, 쿼리 및 구조도)
- 4. 결과
- 5. 참고자료

추천시스템 4

- 1. 추천시스템의 이름
- 2. 추천시스템의 테마 및 목적
- 3. 구현 로직 (코드, 쿼리 및 구조도)
- 4. 결과
- 5. 참고자료

추천시스템 5

- 1. 추천시스템의 이름
- 2. 추천시스템의 테마 및 목적
- 3. 구현 로직 (코드, 쿼리 및 구조도)
- 4. 결과
- 5. 참고자료

회고

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

• 최대 할인율을 적용한 가성비 제품 득템하기

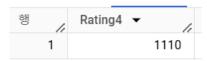
2. 추천시스템의 테마 및 목적

• 할인율이 높으며, 평점 4 이상인 가성비 높은 제품 쇼핑을 제안

3. 구현 로직 (코드, 쿼리 및 구조도)

- 할인율(discount_percentage)을 기준으로 내림차순 정렬
- 평점이 4점 이상인 제품만 포함

select count(*) as Rating4
from amiable-nova-447401-t6.projects.amazon
where safe_cast(rating AS FLOAT64) >= 4;



평점이 4.0인 제품은 총 1110개로, 이를 할인율로 내림차순 했을 때 상위 100개인 제품만 추출하기로 함

select product_id, product_name, category, discounted_price, actual_price, discount_
from amiable-nova-447401-t6.projects.amazon
where safe_cast(rating AS FLOAT64) >= 4
order by discount_percentage DESC limit 100;

4. 결과

• 쿼리 결과



- looker studio
 - https://lookerstudio.google.com/s/ICeQ8oy351w

5. 참고자료

.

추천시스템 2

1. 추천시스템의 이름

유사 구매 고객 기반 상품 추천

2. 추천시스템의 테마 및 목적

- 동일한 카테고리의 제품을 구매한 다른 고객들이 구매한 또 다른 제품을 추천
- 즉, 같은 카테고리의 제품을 구매한 고객들이 추가로 구매한 제품 추천

3. 구현 로직 (코드, 쿼리 및 구조도)

- 1. Value CTE: 카테고리와 사용자 이름 추출
- 2. user_purchases CTE: 사용자 별(name_value) 구매 제품(product_name) 저장
- 3. **frequent_purchases** CTE: 동일 사용자(user1 or user2)가 구매한 다른 제품을 JOIN하고 구매 횟수를 COUNT
- 4. **최종**: 특정 제품을 구매한 사용자가 가장 많이 구매한 다른 제품 추천 (purchase_count를 DESC로 정렬하여 상위 추천 제품 정렬)

최종 쿼리

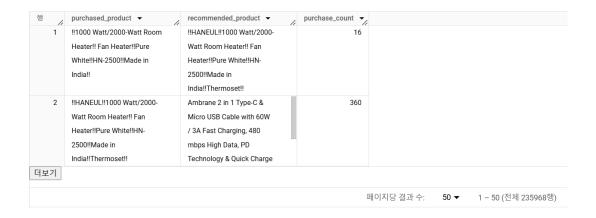
```
WITH Value AS
(
SELECT *
FROM amiable-nova-447401-t6.projects.amazon,
UNNEST(SPLIT(category, '|')) AS category_value,
```

```
UNNEST(SPLIT(user_name, ',')) AS name_value
),
user_purchases AS (
   -- 특정 제품을 구매한 사용자 목록
   SELECT
       name_value,
       product_name AS purchased_product
   FROM Value
),
frequent_purchases AS (
   -- 같은 사용자가 추가로 구매한 제품 찾기
   SELECT
       up1.purchased_product AS base_product, -- 기준 제품
       up2.purchased_product AS recommended_product, -- 추가 구매 제품
       COUNT(*) AS purchase_count
   FROM user_purchases up1
    JOIN user_purchases up2
       ON up1.name_value = up2.name_value -- 같은 사용자
       AND up1.purchased_product <> up2.purchased_product -- 다른 제품
   GROUP BY up1.purchased_product, up2.purchased_product
-- 최종 추천 제품 조회
SELECT
   base_product AS purchased_product,
    recommended_product,
   purchase_count
FROM frequent_purchases
ORDER BY base_product, purchase_count DESC;
```

4. 결과

- 구글 스프레드 시트 필터를 통해 원하는 제품 선택
 - https://docs.google.com/spreadsheets/d/1CynauT_Ylbzg50vhc5GA9IZfl-7luncv7tpY-GduVQQ/edit? gid=1175546894#gid=1175546894

•



5. 참고자료

•

추천시스템 3

1. 추천시스템의 이름

• 카테고리별 베스트 셀러 추천 시스템

2. 추천시스템의 테마 및 목적

• 소비자가 원하는 카테고리 내에서 베스트셀러를 쉽게 찾을 수 있도록 함

3. 구현 로직 (코드, 쿼리 및 구조도)

• 코드

```
WITH category_rank AS (
    SELECT
        category_value,
        product_id,
        product_name,
        rating_count,
        rating,
        RANK() OVER(PARTITION BY category_value ORDER BY rating_count DESC) AS rank
FROM amiable-nova-447401-t6.projects.amazon,
    UNNEST(SPLIT(category, '|')) AS category_value
    WHERE safe_cast(rating AS FLOAT64) >= 4.0
)
```

```
SELECT
   category_value,
   product_id,
   product_name,
   rating_count,
   rating
FROM category_rank
WHERE rank <= 5;</pre>
```

1. category_rank CTE:

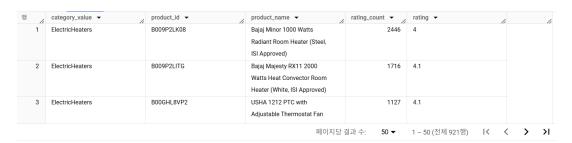
- 각 카테고리별로 rating_count를 기준으로 상품을 내림차순 정렬하고, RANK() 함수로 순위를 매김
- 평점이 4.0 이상인 상품만 필터링

2. 최종 쿼리:

• 상위 5개의 상품만 출력하며, 각 카테고리별로 가장 리뷰가 많은 상품을 추천

4. 결과

• 코드 실행 결과



- google 스프레드 시트 필터를 통해 카테고리 선택
 - https://docs.google.com/spreadsheets/d/1fLXaVGHugo5fIVHoKy94AFMkIJiT8IsbXPluNpBmPZs/edit? usp=sharing

5. 참고자료

_

추천시스템 4

1. 추천시스템의 이름

• 부정적 리뷰 제외 제품 추천

2. 추천시스템의 테마 및 목적

• 부정적 키워드가 포함된 리뷰가 작성된 제품을 제외하고, 긍정적 리뷰의 제품만 추천하기 위함

3. 구현 로직 (코드, 쿼리 및 구조도)

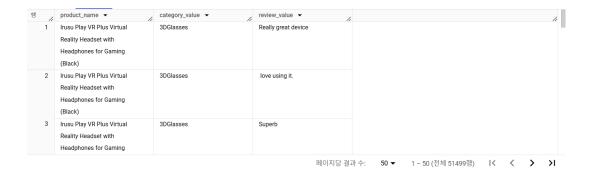
• 코드

```
WITH Value AS (
    SELECT a.*,
           category AS category_value,
           review_title AS review_value
    FROM `amiable-nova-447401-t6.projects.amazon` AS a
    CROSS JOIN UNNEST(SPLIT(a.category, "|")) AS category
    CROSS JOIN UNNEST(SPLIT(a.review_title, ",")) AS review_title
), ReviewFlag as(
  SELECT
    product_name,
    category_value,
    review_value,
    CASE
        WHEN CONTAINS_SUBSTR(review_value, 'not good') or CONTAINS_SUBSTR(review_val
        or CONTAINS_SUBSTR(review_value, "don't buy") or CONTAINS_SUBSTR(review_valu
        THEN 1
        ELSE 0
    END AS negative_review_flag
FROM Value
select product_name,
    category_value,
    review_value from ReviewFlag where negative_review_flag = 0
ORDER BY category_value;
```

- 1. Value CTE: category와 review_title을 분할하여 각 카테고리와 리뷰 제목을 별도로 추출
- 2. ReviewFlag CTE: 리뷰 제목에 not good, fake, don't buy, bad 등의 키워드가 포함된 리뷰를 부정적 리뷰로 표시
- 3. 최종: 부정적 키워드가 포함되지 않은 리뷰만 출력하고, 카테고리별로 정렬하여 반환

4. 결과

• 쿼리 결과



looker studio

https://lookerstudio.google.com/s/p_MI54b5sb8

5. 참고자료

•

추천시스템 5

1. 추천시스템의 이름

•

2. 추천시스템의 테마 및 목적

•

3. 구현 로직 (코드, 쿼리 및 구조도)

•

4. 결과

•

5. 참고자료

•

회고

사용자에게 추천 제품을 보여줄 때, product_name와 product_id 중에 어떤 걸로 표시하는 것이 좋을 지 모르겠다. looker studio로 원하는 결과물을 만들 수 있을 때까지 연습해야겠다!