



# 테스트 계획 및 결과서

## LLM 답변 테스트

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 시간 (초)	토큰 사용량 (입력/출력)
LLM-I-001	인터뷰어가 같은 질문을 반복하지 않는지 확인	사용자 입력 "20대 여성에게 선물할 향수를 추천해줘" 입력 → 이미지 정보가 없기 때문에 LLM 질문 → 사용자가 "잘 모르겠어"라고 대답 → Interviewer가 이전과는 다른 질문을 생성하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-4o-mini	"20대 여성에게 선물할 향수를 추천해줘", "잘 모르겠어"	"선물하시려는 분의 이미지가 차가운 편인가요? 아니면 따뜻한 편인가요?", "그럼 선물하시려는 분의 성격을 대략적으로 알려주세요. (시크한편, 귀여운편 등)"	"평소에 어떤 스타일이나 이미지를 선호하는지 알려주시면 더 구체적으로 추천해드릴 수 있어요. 예를 들어, 자연스러운 느낌인지, 세련된 느낌인지 말씀해 주시면 좋습니다.", "20대 여성에게 어울리는 향을 추천해드릴게요. 평소 선호하는 향이나 스타일이 있나요? 아니면 계절별 향 추천을 원하시나요?"	사용자가 잘 모르겠다고 했을때 다른 방향에서의 이미지 접근이나 계절을 제시하는 모습을 보임  - 같은 질문을 반복하지 않는 것은 성공했으나 때때로 해당 향수 추천의 목적을 제대로 파악하지 못하는 모습을 보임(예시 : 선물할 향수를 추천해달라고 했는데 선물용인지 본인 사용목적인지 질문) 상시적인 내용은 아니기 때문에 프롬핑으로 목적을 파악하고 재질문 하지 않도록 프롬핑 예정	2.43s + 2.93s = 5.36s	936/92 + 1549/188
LLM-R-001	사용자가 특정 브랜드를 추천을 제한했을 경우 제한된 요소에서 검색했거나 검색하지 못했을 경우 왜 그 브랜드에서 추천하지 못했는지 설명하는지 확인	사용자 입력 "비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 사넬 향수 추천해줘" 입력 → 사넬 / 비즈니스 키워드 추출 → 사넬 향수중에 비즈니스 키워드에 맞는 향수 추천	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 사넬 향수 추천해줘"	"사넬향수 1, 사넬향수 2, 사넬향수 3"형태로 모든 향수를 사넬 브랜드의 향수에서 추천	"Chanel - Bleu de Chanel, Chanel - N°18, Chanel - N°19"향수 추천 모든 브랜드가 사넬임을 확인	브랜드 필터링 조건은 강력하게 적용하는 것을 확인  - 하지만 특정 향수를 기준으로 할 경우 Interviewer가 분위기에 대한 정보 부족으로 인식하고 재질문 확인 이후 키워드에서도 향수 이름기준의 필터링이 부족한것으로 확인 검색 로직에서 브랜드와 같은 방식으로 검색하도록 로직을 수정 예정	24.42s	
LLM-R-002	사용자의 분위기/이미지/상황에 따라서 3가지의 다른 전략으로 구성된 향수를 각 전략당 1개씩 추천해주는지 확인	사용자 입력 "차가운 이미지의 남자에게 어울리는 향수 추천해줘" 입력 → 사용자의 입력에서 "차가운"이미지 추출 → 차가운 느낌의 향수, 반전으로 차가운 사람에게도 어울리는 따뜻한 향수, 처음에는 차가운 느낌이지만 점점 따뜻해지는 향수 등의 3가지 다른 전략으로 추천	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"차가운 이미지의 남자에게 어울리는 향수 추천해줘"	"차가운 향수, 차가운 이미지에 어울리는 다른 느낌의 향수, 차가운 이미지에 의외로 잘어울리는 따뜻한 향수"를 추천	"[얼음처럼 차가운 청량한 물향] Kenzo - L'Eau par Kenzo pour Homme, [차가운 수트에 어울리는 드라이 우디] Givenchy - Cuir Blanc, [차가운 얼굴에 의외로 설레는 은은한 비누향] Montale - Aqua Gold"	차가운 이미지라는 키워드에서 3가지 동조, 반대, 의외의 조합 등의 전략을 수립하고 추천하는 것을 확인	31.35s	
LLM-W-001	검색 결과를 기반으로 생성하는지 확인	사용자 입력 "20대 시크한 여성에게 어울리는 향수 추천해줘" 입력 → 시크함에 맞춰서 3가지 키워드로 검색 → 결과로 나오는 향	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"20대 시크한 여성에게 어울리는 향수 추천해줘"	내부적으로 로그로 Researcher에서 검색된 내용에 결과 향수가 모두 포함되어 있는지 확인	no.1, 브랜드: Moschino, 이름: Fresh Couture/no.5, 브랜드: Givenchy, 이름: Irrésistible	출력에서 나온 정보들과 모두 대응되는 내용임을 확인  - 노트 정보의 경우 해당 노트를 노	33.40s	

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 시간 (초)	토큰 사용량 (입력/출력)
		수의 정보들과 해당 검색 결과가 동일한지 확인				Givenchy/no.1, 브랜드: Parfums de Marly, 이름: Kuhuyan	트 이름 없이 최대한 풀어서 전달하라고 명령했기 때문에 명확한지는 판단이 주관적임 RAG 평가와 함께 추가적인 평가지표를 설정할 필요성이 보임		
LLM-W-002	검색 결과가 없을 경우 DB에 없는 함수를 만들어내어 추천하지 않는지 확인	<p>항수Researcher에서 강제로 return {"research_result": "검색 결과가 없습니다", "route": "writer"}를 전달 → 이후 Writer가 가져정보를 출력하는지 확인</p>	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	“여름에 남자가 쓸 상큼한 함수 추천해줘”(의미 없는 입력)	“죄송합니다 현재 제가 가진 정보에서는 해당하는 함수를 찾을 수 없었어요. 혹시 다른 분위기나 조건을 제시해 주실 수 있을까요? ~~ 느낌은 어떠신가요?”의 형태로 출력	<p>“찾으시는 조건에 딱 맞는 함수가 없습니다.아마도 DB에 **여름/남성/상큼**처럼 사용하신 조건(계절, 성별, 분위기)을 동시에 만족하는 함수가 <b>등록되어 있지 않거나</b>, 참조 태그가 <b>시트러스·아쿠아·그린</b>처럼 더 세분화되어 있어서 “상큼”이라는 표현으로는 매칭이 안 됐을 가능성이 큼니다. 또 <b>남성 함수</b>만으로 제한되면서 결과가 더 줄었을 수도 있어요.대신 비슷한 방향으로 추천을 도와드릴게요. 아래 중에서 어떤 쪽을 원하시나요?”상큼“이 <b>시트러스(레몬/자몽)</b> 느낌인가요, 아니면 <b>아쿠아(바다/비누)</b> 쪽인가요, 혹은 <b>그린(허브/풀)</b> 쪽인가요? <b>가격대</b>(예: 5만/10만/20만 원대)와 <b>지속력 선호</b>(가벼움 vs 오래감)가 있으신가요?<b>브랜드 제한</b>(예: 니치/디자이너, 특정 브랜드) 있으세요?이 답변 주시면, DB 조건을 조금 넓히거나 (예: “시트러스/아쿠아/그린”으로) 비슷한 무드로 다시 찾아서 추천해드릴게요.”</p>	<p>없는 정보를 만들어서 추천하지는 않는 것을 확인</p> <p>- 전문적인 아쿠아, 그린, 시트러스들의 표현을 사용하지만 옆에 예시를 알기 쉽게 써서 큰 문제가 보이지는 않음</p> <p>- 하지만 제3문의 내용이 너무 방대하여 추가적인 프롬프팅으로 예시를 제시하는 개수를 줄여야할 필요성을 발견</p>	7.82s	
LLM-W-003	각 전략당 1개의 함수를 추천하는지 확인	<p>사용자 입력 “여름에 여자가 쓸 상큼한 함수 추천해줘” 입력 → 여름&amp;상큼을 키워드로 3가지 전략을 수립 → 각 전략에 나온 함수들이 1개씩 추천되는지 확인</p>	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	“여름에 여자가 쓸 상큼한 함수 추천해줘”	“여름에 맞는 상큼한 향, 상큼함 속에 포근한 향, 여름을 날려버릴 시원한 향”등의 3가지 전략 각각에 1개의 함수를 추천	<p>“[여름 과일물 같은 상큼 시트러스] Estee Lauder - Pure White Linen Pink Coral, [상큼한데 은근 꽃향기 나는 반전] Jo Malone London - White Moss &amp; Snowdrop, [바다 바람 같은 상큼-청량 밸런스] Calvin Klein - CK One Summer 2015”</p>	<p>예상 했던것 보다 사용자의 입력에서 상큼함에 더 집중하여 더 좋은 전략을 제시하는 것을 확인 또한 검색 결과와 비교하여 각 전략별로 하나씩 제시하는 것 또한 확인</p>	32.52s	

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 시간 (초)	토큰 사용량 (입력/출력)
LLM-W-004	적용된 서식을 잘 지키는지 (목록표시, 구분선표시, 색상지정등)	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 추천되어 나오는 결과 향수가 이미지가 센터크롭 되어있고 목차별로 구분되어있고 구분선이 표시되고 향표현등의 강조할 내용에 색표현이 들어가는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	<b>1.[전략]브랜드-향수명 이미지 어떤 향인가요?:</b> <b>내용</b> <b>추천 이유:</b> 내용 <b>정보:</b> 브랜드 / 향수명 / 출시년도	<b>1. [여름 과일들 같은 상큼 시트러스]</b> 이미지 <b>Estee Lauder - Pure White Linen Pink Coral 어떤 향인가요?:</b> 처음엔 잘 익은 빨간 과일과 특쓰는 후추 한 방울처럼 경쾌하게 터져요. 곧 <b>벚꽃-자스민</b> 같은 <b>여리여리한 꽃향기</b> 가 올라오면서, 마지막엔 <b>부드러운 나무와 은은한 바닐라</b> 가 살짝 포근하게 정리해줘요. <b>추천 이유:</b> "여름에 여자"라는 조건에서 가장 중요한 건 <b>덥고 습한 날에도 답답하지 않게 산뜻한 인상</b> 이에요. 이 향은 <b>상큼함이 먼저 와서 시원하고</b> , 뒤에 <b>가벼운 꽃향기</b> 로 여성스럽게 마무리돼서 데일리로 실패 확률이 낮아요. <b>정보:</b> Estee Lauder / Pure White Linen Pink Coral / 출시년도 정보 없음	주어진 형식에 맞게 잘 출력하는 것을 확인 이미지 센터 크롭 적용, 목차별 구분과 구분선 표시, 향 표현등의 강조할 표현에 색표현과 볼드로 강조를 하되 어떤향인가요 같은 소재목과는 다른 색으로 표현	32.52s	
LLM-W-005	항을 설명할때 전문적인 용어를 사용하지 않는지	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 추천 설명 전반에 플로럴, 노트, 탑/미들/베이스 등의 전문적이 표현이 포함되지 않는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	<b>어떤 향인가요?:</b> 처음엔 <b>과일속에서 터지는 후추 한 방울</b> 처럼 경쾌하게 터져요. 곧 <b>자스민의 꽃향기가 부드럽게</b> 올라오면서, 마지막엔 <b>부드러운 나무향</b> 으로 포근하게 정리해줘요.	<b>어떤 향인가요?:</b> 시작은 <b>귤껍질들 살짝 비틀어 툭 터뜨린 듯한 상큼함</b> 이 먼저 와요. 시간이 지나면 <b>하얀 꽃의 깨끗한 향</b> 이 조용히 올라오고, 끝에는 <b>이끼 낀 숲 바닥 같은 보송보송한 초록 기운</b> 과 <b>은근한 달콤함</b> 이 피부에 얇게 남아요.	실제로 노트나 어코드 같은 전문적인 표현을 직접적으로 사용하지 않고 사용자가 이해하기 쉬운 표현으로 치환하여 표시하고 있는 것을 확인  - 머스크, 레몬등의 일상적으로 사용하게 되는 표현의 경우 사용이 허용되는데 머스크 등의 일부 표현의 경우 대중적이다의 기준이 모호하여 정의가 필요함	32.52s	
LLM-W-006	추천의 이유를 사용자의 쿼리와 잘 이어서 제시하는지	사용자 입력 "20대 여성에게 선물할 향수를 추천해줘" 입력 → 재질문 → 사용자 입력 "귀여운 이미지" 입력 → 추천의 이유를 설명할때 20대 여성에게 선물할 목적, 귀여운 이미지에 맞는 키워드가 선택되었는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"20대 여성에게 선물할 향수를 추천해줘", "귀여운 이미지"	<b>추천 이유:</b> 귀여운 20대 여성분에게는 상큼하고 달콤한 향이 잘 어울립니다. 그래서 새콤달콤한 과일향을 가진 향수를 추천해드렸어요.	<b>추천 이유:</b> "귀여운" 이미지를 가장 <b>직관적</b> 으로 만들기 쉬운 타임이에요요. 20대 여성에게 잘 어울리는 <b>상큼달콤 과일 사탕</b> 느낌이라, 향수 초보도 실패 확률이 낮아요.	추천이유를 사용자의 쿼리와 잘 연결지어서 설명  - 이전 프롬프트에서는 20대의 여성과 같은 대상에 관한 직접적인 언급이 들어갔었는데 프롬프트 수정과 정에서 결과도 함께 바뀐 것으로 추정 이미지뿐만 아니라 해당 대상에 관한 분석과 이유	3.23s + 26.82s = 30.05s	

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 시간 (초)	토큰 사용량 (입력/출력)
							를 제시하도록 프롬프트 강화 예정		
LLM-W-007	향의 변화를 탐/미들/베이스 순으로 시간의 흐름에 따라 설명하는지	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 향의 표현을 설명하는 부분에 ~~느낌으로 시작되어 시간이 좀 지나면 ~~느낌이 강해지고 ~~향이 은은하게 남습니다. 의 형태로 표현되는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	<b>어떤 향인가요?:</b> 처음엔 과일속에 서 터지는 후추 한 방울처럼 경쾌하게 터져요. 곧 자스민의 꽃향기가 부드럽게 올라오면서, 마지막엔 부드러운 나무향으로 포근하게 정리해줘요.	<b>어떤 향인가요?:</b> 시작은 자몽처럼 산뜻한 과일 향이 가볍게 튀고, 이어서 벗꽃-라일락-작약 같은 꽃 향이 비누 거품처럼 뽀얗고 깔끔하게 퍼져요. 마지막엔 하얀 머스크가 남아 막 시원한 사람 같은 청결감이 오래가요.	실제로 탐/미들/베이스 노트가 각각 [Top]: Currant, Passion fruit, Pink grapefruit [Middle]: Cherry blossom, Lilac, Peony, Rose [Base]: Cedar, White musk 로 각각의 요소를 시간 순서대로 정리하여 제시  - 여러 요소가 섞여있는 경우 특정 향들로 줄이는 경향성이 있는 것으로 파악되어 제약 조건에 관한 논의 필요	26.82s	
LLM-W-008	전혀 상관없는 이야기가 들어왔을 때 잘 대처하는지 (필요에 따라 별도의 노드로 구성)	사용자 입력 "오늘 날씨 어때?" → 응답"아쉽지만 저는 날씨를 알 수 없습니다. 혹시 오늘은 따뜻한가요? 아니면 쌀쌀한가요? 알려주시면 오늘에 어울리는 향수를 추천해드릴게요!"의 형식의 답변을 출력	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"오늘 날씨 어때?"	"아쉽지만 저는 날씨를 알 수 없습니다. 혹시 오늘은 따뜻한가요? 아니면 쌀쌀한가요? 알려주시면 오늘에 어울리는 향수를 추천해드릴게요!"	앗, 저는 서버 안에서만 살아서 오늘 바깥 날씨를 실시간으로 못 봤어요—미안해요. 😊 지금 계산 곳은 맑아요, 흐려요, 비와요?	기본적인 내용들은 대답이 가능하지만 출력 형식이 앳, 저는 서버에만 살아서 ~~ 형태로 제시되는 느낌이 강해서 좀 더 자연스러운 대화가 가능하도록 프롬프트 수정 예정	2.46s	360/6

## 로직 테스트

TEST-ID	테스트 목적	테스트 환경	테스트 과정	기대 상황	실제 상황	분석 및 개선점
Logic-S-001	Supervisor가 이미지/분위기에 연관된 정보의 부재를 잘 포착하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-4o-mini	사용자 입력"20대 여성에게 선물할 향수 추천해줘"입력 → 명확한 이미지나 분위기에 관한 정보 부재를 인식 → 내부 로그를 통해 Interviewer에게 전달되는지 확인	이미지 정보의 부재를 인식하여 로그에 표시	 [Supervisor] 입력: '20대 여성에게 선물할 향수 추천해줘'  결정된 경로: interviewer	분위기 정보의 누락하고 Interviewer
Logic-I-001	Interviewer가 이미지/분위기에 연관된 정보의 부재를 잘 포착하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-4o-mini	사용자 입력"20대 여성에게 선물할 향수 추천해줘"입력 → 명확한 이미지나 분위기에 관한 정보 부재를 인식 → 이미지/분위기에 관련된 질문을 제시	이미지 정보의 부재를 인식하고 이미지 관련 질문을 출력	"20대 여성분이시군요! 평소 어떤 스타일(러블리, 쿨함 등)이나 찾으시는 분위기가 있으신가요?"출력	해당 인물의 스타일에 관한 질문을 ? Supervisor와 함께 판단함을 확인
Logic-R-001	검색SQL 구문 작성을 위한 요소를 영어로 Tool에 전달하고 있는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"시크한 남자한테 선물할 향수 추천해줘" 입력 → 바로 시크한, 남자의 키워드를 이용해 Tool에 전달하는지 확인	남자의 키워드는 gender에 포함되어 전달하고 시크한은 매칭되는 accord와 시크함을 검색하여 나온 note정보를 전달	[Priority 1] 실행: 블랙 수트 같은 우디 [Step 1] Strict: [{'column': 'gender', 'value': 'Masculine'}, {'column': 'accord', 'value': 'Woody'}, {'column': 'note', 'value': ['Burnt vetiver', 'Shangri-Lapeony', 'Atlas cedar Orpur', 'Bourbon vetiver', 'Soil', 'Lorenox', 'Cream', 'Salt', 'Chili', 'Paeonia mascula', 'Bran', 'Atlas cedar', 'Dry notes', 'Cep', 'Minerals', 'White musk', 'Dry woods', 'Soap']}]	검색에 남성의 키워드 시크함에 어울리 Woody 어코드를  - 노트 정보에 대별도의 평가지표 보유 RAG 평가5행될 예정
Logic-R-002	명확하게 사용자의 쿼리에 포함된 내용을 SQL 쿼리에 우선 포함 시키는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	사용자 입력 "비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 샤넬 향수 추천해줘" 입력 → SQL 쿼리문에 Brand : Chanel이 포함되는지 확인	모든 향수 SQL 쿼리문에 brand Chanel이 포함됨	[Priority 1] 실행: 비누처럼 깔끔한 프렌치 셔츠 냄새  [Step 1] Strict: [{'column': 'brand', 'value': 'Chanel'}]  [Priority 2] 실행: 깔끔한데 은근히 우디한 신뢰감  [Step 1] Strict: [{'column': 'brand', 'value': 'Chanel'}]  [Priority 3] 실행: 차가운 프레시함에 살짝 스파이시한 반전	모든 전략에 Chanel 포함됨 확인  - 하지만 검색결: 경우 어떤 것이 4되는지는 확인해

TEST-ID	테스트 목적	테스트 환경	테스트 과정	기대 상황	실제 상황	분석 및 개선점
					🔍 [Step 1] Strict: [{ 'column': 'brand', 'value': 'Chanel' }]	
Logic-T-001	벡터검색을 위해서 OpenAI의 text-embedding-3-small을 이용해서 구문을 잘 임베딩 하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector)	특정 쿼리를 입력했을 때 SQL 문에 들어갈 Note정보를 잘 가져오는지 확인 Lemon이라는 정보를 넣었을 때 Note에 Lemon, Bergamot이 포함되는지를 확인	Note에 Lemon, Bergamot 등의 레몬/귤 계열의 노트를 포함	Strict: [{ 'column': 'gender', 'value': 'Feminine' }, { 'column': 'accord', 'value': 'Citrus' }, { 'column': 'note', 'value': [ 'Bulgarian rose', 'Tiaré', 'Privet', 'Cream', 'Prosecco', 'Paeonia mascula', 'Lemon zest', 'Gin fizz', 'Fresh notes', 'Amalfi lemon', 'Italian lemon zest', 'Blue sky accord', 'Bitter lemon' ] }]	전락은 Citrus, Lemon, Italian의 레몬 계열 Note하고 있는것을 확인  - 하지만 Bulgaria 등의 직접적으로 어보이는 요소에 Lemon계열인지인 검증이 필요
Logic-T-002	검색된 향수의 정보를 전달하기 위해서 투표수에 기반해서 적은 득표를 한 요소를 잘 제거하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector)	사용자 입력 "비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 샤넬 향수 추천해줘" 입력 → 추천되어 나온 향수의 모든 Accord, Season, Occasion 득표와 실제 전달된 Json에서 전달된 Accord, Season, Occasion의 표가 잘 제거되었는지 확인	해당 영역에서의 투표수가 10% 이하의 경우 특징이 아닌것으로 판단하여 해당 요소 삭제 처리	<b>Accord</b> Fresh:90, Citrus:79, Floral:65, Fruity:43, Woody:30, Green:25 <b>Synthetic:</b> 19 <b>Occasion</b> Daily:94, Leisure:88, Business:87, Sport:41, Evening:26, Night Out:24 <b>Season</b> Summer:94, Spring:91, Fall:23, Winter:11 =====실제 Json===== 분위기: Citrus, Floral, Fresh, Fruity 계절: Spring, Summer 상황: Business, Daily, Leisure, Sport	10% 이하의 요소적으로 삭제 처리
Logic-T-003	향수의 노트에 대한 결과를 반환할 때 Top/Middle/Base에 따라서 구분하여 Json 형식으로 반환해주는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector)	사용자 입력 "비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 샤넬 향수 추천해줘" 입력 → Researcher에서 Writer로 넘어갈 때 Top/Middle/Base를 구분하여 넘어가는지 확인	Langsmith에서 Researcher에서 Writer로 넘어갈 때 Top/Middle/Base를 구분하여 넘어가는지 확인	no.2 브랜드: Chanel 이름: Chance Eau Vive 이미지: <a href="https://media.parfumo.com/perfume_social/ad/ad03ca-chance-eau-vive-eau-de-toilette-chanel-1200.jpg?format=jpg&amp;quality=90">https://media.parfumo.com/perfume_social/ad/ad03ca-chance-eau-vive-eau-de-toilette-chanel-1200.jpg?format=jpg&amp;quality=90</a> 성별: Classic, Feminine, Modern 분위기: Citrus, Floral, Fresh, Fruity 계절: Spring, Summer 상황: Business, Daily, Leisure, Sport 노트 구성: [Top]: Blood orange, Grapefruit [Middle]: Jasmine [Base]: Cedar, Iris	정상적으로 노트 Top/Middle/Base되어 넘어가는 것
Logic-T-004	사용자가 명확한 브랜드명, 향수명을 제시했을 때 SQL 쿼리에 브랜드명, 향수명이 포함되는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector)	사용자 입력 "샤넬의 Chance Eau Vive와 비슷한 향수 추천해줘" 입력 → search_exact_entity 함수에서 반환하는 값에 Chanel, Chance Eau Vive가 포함되는지 확인	내부 로그를 조회하여 Chanel, Chance Eau Vive가 SQL 쿼리에 포함되는지 확인	SELECT perfume_brand FROM tb_perfume_basic_m WHERE perfume_brand ILIKE %s LIMIT 1 ('Chanel',)	브랜드 정보의 관계 범주명을 RDI하여 일치시 SQL함  - 향수명의 경우 상태 수정 필요 - 해당 형식의 질 Interviewer나 Supervisor가 ?으로 판단 프롬프 필요
Logic-T-005	note 검색시 (Hybrid 서치 적용 여부) 1. RDB의 범주명과 정확히 일치시 사용 2. RDB의 범주명과 오탈타 검사후 3글자 이하의 경우 사용 3. VectorDB의 분위기/이미지 임베딩에서 검색 후 사용	Docker Container (Python 3.11), PostgreSQL 17 (pgvector)	사용자 입력 "레몬 들어간 향수추천해줘" 입력 사용자 입력 "Lemon 들어간 향수" 입력 사용자 입력 "상큼한 향수" 입력 각각의 결과가 1,2,3에서 작동하는지 여부 확인	내부 로그를 조회하여 각 사용자 입력이 1,2,3으로 필터링되어 알맞은 방식으로 적용되는지 확인	# 1 SELECT note FROM tb_note_embedding_m WHERE note ILIKE %s LIMIT %s %Lemon% # 3 SELECT note FROM tb_note_embedding_m WHERE 1=1 ORDER BY embedding <=> %s::vector LIMIT %s;	1번 케이스에 대해 상적으로 작동들 2번 케이스의 경가 분리되지 않고 - 브랜드: 정보 없음 - 이미지: 시크 - 취향/향조: Larmon) - 상황(TPO): 정 - 대상(성별): 남의도 반영) 의 전체 형태로 1것을 확인 수정 <3번도 정상작동>지만 내부적으로 분리되지 않은 등딩 된것으로 확인워드 쿼리를 정규 분리 예정
Logic-T-006	각 조건문에서 note, accord 등	Docker Container	사용자 입력 "비즈니스 미팅에 뿌리고 갈 깔끔	Note<=>Accord<=>Brand 등의 요소에 대해서는	[WHERE 절]: 1=1 AND b.perfume_brand IN (%s) AND a.audience IN (%s) AND o.occasion IN (%s) AND	Note<=>Accord 등의 요소에 대해

