



Episode 2: 정보 탐험대 — SQL 쿼리의 여정

도시 곳곳의 정보를 탐색하는 SQL 전문가들의 여정

정보 탐험대는 도시 곳곳에 숨겨진 정보를 찾아내는 전문가들이었어요.

그들의 무기는 바로 **SQL 쿼리**!

이들은 **SELECT** 주문서를 들고, **WHERE** 조건을 붙이고, **JOIN** 마법으로 연결하며,

GROUP BY로 정보를 묶고, 윈도우 함수로 시간의 흐름을 분석했죠.

왜 필요한가?

SQL은 데이터베이스와 대화하는 언어예요.

- 정보를 꺼내고, 수정하고, 삭제하고, 추가하는 모든 작업은 SQL로 이루어져요.
- SQL을 잘 다루면, 데이터 도시의 모든 비밀을 자유롭게 탐험할 수 있어요.

탐험대의 기술 요약

기술	설명	예시
SELECT	원하는 정보 조회	'이름과 주소 알려줘!'
WHERE	조건 필터링	'서울에 사는 사람만 보여줘!'
JOIN	테이블 연결	'고객 정보와 주문 내역을 연결해줘!'
GROUP BY + HAVING	그룹화 + 조건	'지역별로 주문 수를 세고, 10건 이상인 곳만 보여줘!'
서브쿼리	쿼리 안의 쿼리	IN, EXISTS, NOT EXISTS
스칼라 서브쿼리	단일 값 반환	SELECT 이름, (SELECT 부서명 FROM 부서 ...)
윈도우 함수	행별 계산	ROW_NUMBER(), RANK(), DENSE_RANK()
OVER()	그룹 내 순서 지정	OVER(PARTITION BY ... ORDER BY ...)

🔗 SQL 문장 구조 vs 실행 순서

구분	순서
문장 구조	SELECT → FROM → WHERE → GROUP BY → HAVING → ORDER BY
실행 순서	FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

💡 **알기 포인트:** 실행은 FROM부터 시작!

🔗 JOIN 유형 구분

JOIN 유형	설명	예시
INNER JOIN	공통된 값만 조회	고객과 주문이 모두 있는 경우
LEFT JOIN	왼쪽 테이블은 모두, 오른쪽은 조건 일치 시	고객은 모두, 주문이 없어도 표시
RIGHT JOIN	오른쪽 테이블은 모두	주문은 모두, 고객이 없어도 표시
FULL OUTER JOIN	양쪽 모두 표시	고객/주문 모두 표시, NULL 포함

```
SELECT *
FROM 고객
JOIN 주문 ON 고객.고객ID = 주문.고객ID;
```

✂ 서브쿼리 유형별 구분

서브쿼리 유형	설명	예시
스칼라	단일 값 반환	SELECT 이름, (SELECT 부서명 ...)
인라인 뷰	FROM 절에 사용	SELECT * FROM (SELECT ...) AS T
다중행	여러 값 반환	WHERE 컬럼 IN (SELECT ...)
상관 서브쿼리	외부 쿼리와 연결	WHERE EXISTS (SELECT ... WHERE 외부.컬럼 = 내부.컬럼)

💡 **알기 포인트:** 스칼라 = 하나 / 인라인 = 테이블처럼 / 상관 = 외부 연결

💡 윈도우 함수 실전 예시

SELECT 이름, RANK() OVER(PARTITION BY 부서 ORDER BY 급여 DESC) AS 순위
FROM 사원;

비교 항목	GROUP BY	윈도우 함수
집계 대상	그룹 전체	각 행 기준
결과 행 수	그룹 수	원래 행 수 유지

💡 **알기 포인트:** 윈도우 함수 = 행별 분석 / GROUP BY = 묶어서 분석

🔧 DML vs DDL 비교

구분	명령어	설명
DML	INSERT, UPDATE, DELETE	데이터 조작 (행 단위)
DDL	CREATE, ALTER, DROP	구조 변경 (테이블, 컬럼 등)

💡 **알기 포인트:** DML = 행 / DDL = 구조

📌 기억하기 쉬운 요약 정리

- **SELECT:** 꺼내기
- **WHERE:** 필터링
- **JOIN:** 연결
- **GROUP BY:** 묶기
- **HAVING:** 그룹 조건
- **서브쿼리:** 쿼리 안의 쿼리
- **윈도우 함수:** 행별 분석
- **DML/DDL:** 데이터 조작/구조 변경
- **트랜잭션:** 작업 묶음 관리

정보 탈출대 암기카드

카드 21

앞면: SELECT 문의 역할은?

뒷면: 원하는 정보 조회

설명: 테이블에서 특정 컬럼을 꺼냄


 암기 포인트: 'SELECT = 꺼내기'

카드 22

앞면: WHERE 절의 기능은?

뒷면: 조건 필터링

설명: 특정 조건에 맞는 행만 조회

 암기 포인트: 'WHERE = 필터링'

카드 23

앞면: JOIN의 목적은?

뒷면: 테이블 연결

설명: 두 테이블의 관련 데이터를 합침

 암기 포인트: 'JOIN = 연결'

카드 24

앞면: GROUP BY의 기능은?

뒷면: 그룹화

설명: 같은 값을 가진 행들을 묶음

 암기 포인트: 'GROUP BY = 묶기'

카드 25

앞면: HAVING 절은 언제 사용하나요?

뒷면: 그룹 조건 필터링

설명: GROUP BY 이후 조건 지정


 암기 포인트: 'HAVING = 그룹 필터'

카드 26

앞면: 서브쿼리란?

뒷면: 쿼리 안의 쿼리

설명: SELECT, WHERE, FROM 등에서 사용


 암기 포인트: '서브쿼리 = 쿼리 속 쿼리'

카드 27

앞면: 스칼라 서브쿼리란?

뒷면: 단일 값 반환

설명: SELECT 절에서 사용 가능

 암기 포인트: '스칼라 = 하나의 값'

카드 28

앞면: IN과 EXISTS의 차이는?

뒷면: IN은 값 비교, EXISTS는 존재 여부 확인

설명: EXISTS는 성능이 더 좋을 수 있음

 암기 포인트: 'EXISTS = 존재 확인'

카드 29

앞면: 윈도우 함수란?

뒷면: 행별 계산 함수

설명: GROUP 없이도 집계 가능


 암기 포인트: '윈도우 함수 = 행별 분석'

카드 30

앞면: RANK와 DENSE_RANK의 차이는?

뒷면: RANK는 건너뛰고, DENSE_RANK는 연속

설명: 동점 처리 방식이 다름


 암기 포인트: 'RANK = 건너뛰기 / DENSE = 연속'

카드 31

앞면: ROW_NUMBER 함수의 역할은?

뒷면: 행 번호 부여

설명: 순서대로 번호 매김


 암기 포인트: 'ROW_NUMBER = 번호 매기기'

카드 32

앞면: OVER 절의 기능은?

뒷면: 윈도우 함수의 범위 지정

설명: PARTITION BY, ORDER BY 포함

 암기 포인트: 'OVER = 분석 범위'

카드 33

앞면: COUNT(*)는 무엇을 세나요?

뒷면: 전체 행 수

설명: NULL 포함

 암기 포인트: 'COUNT(*) = 전체 행'

카드 34

앞면: COALESCE 함수의 역할은?

뒷면: NULL 대체

설명: NULL이면 지정한 값으로 대체


 암기 포인트: 'COALESCE = NULL 처리'

카드 35

앞면: INSERT 문의 기능은?

뒷면: 데이터 추가

설명: 테이블에 새 행 삽입


 암기 포인트: 'INSERT = 추가'

카드 36

앞면: UPDATE 문의 기능은?

뒷면: 데이터 수정

설명: 기존 행의 값을 변경


 암기 포인트: 'UPDATE = 수정'

카드 37

앞면: DELETE 문의 기능은?

뒷면: 데이터 삭제

설명: 조건에 맞는 행 제거


 암기 포인트: 'DELETE = 삭제'

카드 38

앞면: COMMIT의 역할은?

뒷면: 변경 내용 저장

설명: 트랜잭션 완료

 암기 포인트: 'COMMIT = 저장'

카드 39

앞면: ROLLBACK의 역할은?

뒷면: 변경 내용 취소

설명: 트랜잭션 되돌리기


 암기 포인트: 'ROLLBACK = 취소'

카드 40

앞면: DDL 명령어의 종류는?

뒷면: CREATE, ALTER, DROP

설명: 테이블 구조 변경

 암기 포인트: 'DDL = 구조 변경'

카드 41

앞면: JOIN의 유형은?

뒷면: INNER, LEFT, RIGHT, FULL

설명: 테이블 연결 방식에 따라 결과가 달라짐


 암기 포인트: 'INNER = 교집합 / LEFT = 왼쪽 우선 / FULL = 전체'

카드 42

앞면: 서브쿼리의 유형은?

뒷면: 스칼라, 인라인 뷰, 상관 서브쿼리 등

설명: 위치와 반환 방식에 따라 구분됨

 암기 포인트: '스칼라 = 하나 / 인라인 = 테이블처럼 / 상관 = 외부 연결'

카드 43

앞면: 윈도우 함수 실전 예시

뒷면: RANK() OVER(PARTITION BY ... ORDER BY ...)

설명: 그룹 내 순위 계산

 암기 포인트: 'PARTITION BY = 그룹 / ORDER BY = 정렬'

카드 44

앞면: DML과 DDL의 차이는?

뒷면: DML은 행 조작, DDL은 구조 변경

설명: INSERT/UPDATE vs CREATE/ALTER


 암기 포인트: 'DML = 행 / DDL = 구조'

카드 45

앞면: 상관 서브쿼리란?

뒷면: 외부 쿼리의 컬럼을 참조하는 서브쿼리

설명: EXISTS 조건에서 자주 사용됨

 암기 포인트: '상관 = 외부 컬럼 참조'


대표 기출 문제


문제 1

다음 중 SELECT 문의 기능은?

- ① 테이블 삭제 ② 데이터 조회 ③ 데이터 수정 ④ 트랜잭션 저장

 정답: ②

 해설: SELECT는 원하는 컬럼을 꺼내는 명령어예요.


 암기 포인트: 'SELECT = 꺼내기'

문제 2

GROUP BY와 함께 사용되는 조건 필터링 절은?

- ① WHERE ② HAVING ③ ORDER BY ④ FROM

 정답: ②

 해설: HAVING은 GROUP BY 이후 그룹 조건을 필터링할 때 사용돼요.


 암기 포인트: 'GROUP BY 이후 = HAVING'


문제 3

윈도우 함수에서 순위를 건너뛰는 함수는?

- ① DENSE_RANK ② RANK ③ ROW_NUMBER ④ COUNT

 정답: ②

 해설: RANK는 동점자가 있을 경우 순위를 건너뛰고 다음 순서로 넘어가요.

 암기 포인트: 'RANK = 건너뛰'

문제 4

다음 중 트랜잭션을 저장하는 명령은?

- ① ROLLBACK ② DELETE ③ COMMIT ④ SELECT

✓ 정답: ③

💡 해설: COMMIT은 트랜잭션을 완료하고 변경 내용을 저장하는 명령어예요.

🔍 알기 포인트: 'COMMIT = 저장'

문제 5

서브쿼리에서 단일 값을 반환하는 유형은?

- ① 인라인 뷰 ② 스칼라 서브쿼리 ③ 다중행 서브쿼리 ④ 다중열 서브쿼리

✓ 정답: ②

💡 해설: 스칼라 서브쿼리는 SELECT 절에서 단일 값을 반환할 때 사용돼요.

🔍 알기 포인트: '스칼라 = 하나의 값'

문제 6

다음 중 상관 서브쿼리의 특징은?

- ① 외부 쿼리와 독립적으로 실행된다 ② 단일 값을 반환한다 ③ 외부 쿼리의 컬럼을 참조한다 ④ GROUP BY와 함께 사용된다

✓ 정답: ③

💡 해설: 상관 서브쿼리는 외부 쿼리의 컬럼을 내부 쿼리에서 참조하며 반복적으로 실행돼요.

🔍 알기 포인트: '상관 = 외부 컬럼 참조'