윈도우 함수 개념서

이 문서는 SQLD 시험을 위한 윈도우 함수의 종합적인 개념서입니다. 윈도우 함수의 기본 개념, 종류, PARTITION BY 절, 윈도우절, 고급 활용 예제, 성능 고려사항, 그리고 SQLD 시험 대비 팁을 포함하고 있습니다. 데이터베이스 전문가를 목표로 하는 분들에게 필수적인 내용을 담고 있습니다.

1. 윈도우 함수의 기본 개념

1.1 정의

- 행과 행 간의 관계를 정의하여 결과를 집계하는 함수
- 데이터의 순위 매기기, 누적 합계 등을 수행
- 행마다 결과를 반환하면서 그룹화도 가능

1.2 기본 문법

```
윈도우함수(인자) OVER (
    [PARTITION BY 컬럼,...]
    [ORDER BY 컬럼,...]
    [ROWS or RANGE BETWEEN 시작점 AND 끝점]
)
```

1.3 구성 요소

1. **윈도우함수**: 순위, 집계, 행 순서 등의 함수
2. **PARTITION BY**: 그룹화할 컬럼 지정
3. **ORDER BY**: 순서 지정
4. **WINDOWING**: 윈도우의 범위 지정

2. 윈도우 함수의 종류

2.1 순위 함수

- **ROW_NUMBER()**
  - 순차적인 일련번호 부여
  - 동일 값이어도 다른 번호 부여

```
SELECT 사원명, 급여,
    ROW_NUMBER() OVER (ORDER BY 급여 DESC) as 순위
FROM 사원;
```

- **RANK()**
  - 동일한 값에 동일한 순위 부여
  - 다음 순위는 건너뜀

```
SELECT 사원명, 급여,
    RANK() OVER (ORDER BY 급여 DESC) as 순위
FROM 사원;
```

- **DENSE_RANK()**
  - 동일한 값에 동일한 순위 부여
  - 다음 순위를 건너뛰지 않음

```
SELECT 사원명, 급여,
    DENSE_RANK() OVER (ORDER BY 급여 DESC) as 순위
FROM 사원;
```

2.2 집계 함수

- **SUM()**

```
SELECT 부서명, 사원명, 급여,
    SUM(급여) OVER (PARTITION BY 부서명) as 부서별합계
FROM 사원;
```

- **AVG()**

```
SELECT 부서명, 사원명, 급여,
    AVG(급여) OVER (PARTITION BY 부서명) as 부서별평균
FROM 사원;
```

- **COUNT()**

```
SELECT 부서명, 사원명,
    COUNT(*) OVER (PARTITION BY 부서명) as 부서별인원수
FROM 사원;
```

2.3 행 순서 함수

- **FIRST_VALUE()**
  - 파티션 내 첫 번째 값 반환

```
SELECT 부서명, 사원명, 급여,
    FIRST_VALUE(급여) OVER (
        PARTITION BY 부서명
        ORDER BY 급여 DESC
    ) as 부서내최고급여
FROM 사원;
```

- **LAST_VALUE()**
  - 파티션 내 마지막 값 반환

```
SELECT 부서명, 사원명, 급여,
    LAST_VALUE(급여) OVER (
        PARTITION BY 부서명
        ORDER BY 급여 DESC
        ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
    ) as 부서내최저급여
FROM 사원;
```

- **LAG() / LEAD()**
  - LAG(): 이전 행의 값 참조
  - LEAD(): 다음 행의 값 참조

```
SELECT 사원명, 입사일,
    LAG(입사일) OVER (ORDER BY 입사일) as 이전입사일,
    LEAD(입사일) OVER (ORDER BY 입사일) as 다음입사일
FROM 사원;
```

3. PARTITION BY 절

PARTITION BY 절은 윈도우 함수에서 중요한 역할을 합니다.

3.1 기본 활용

```
-- 부서별, 직급별 급여 순위
SELECT 부서명, 직급, 사원명, 급여,
    RANK() OVER (
        PARTITION BY 부서명, 직급
        ORDER BY 급여 DESC
    ) as 순위
FROM 사원;
```

3.2 다중 파티션

```
-- 부서별, 연도별 평균급여
SELECT 부서명,
    EXTRACT(YEAR FROM 입사일) as 년도,
    AVG(급여) OVER (
        PARTITION BY 부서명,
        EXTRACT(YEAR FROM 입사일)
    ) as 평균급여
FROM 사원;
```

4. 윈도우절 (ROWS, RANGE)

4.1 ROWS 절

- 물리적인 행의 수를 기준으로 윈도우 지정

```
SELECT 사원명, 급여,
    SUM(급여) OVER (
        ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
    ) as 누적급여
FROM 사원;
```

4.2 RANGE 절

- 논리적인 값의 범위를 기준으로 윈도우 지정

```
SELECT 사원명, 급여,
    SUM(급여) OVER (
        ORDER BY 급여
        RANGE BETWEEN 1000 PRECEDING AND 1000 FOLLOWING
    ) as 범위급여합계
FROM 사원;
```

4.3 주요 윈도우 지정 옵션

1. **UNBOUNDED PRECEDING**: 파티션의 첫 행
2. **UNBOUNDED FOLLOWING**: 파티션의 마지막 행
3. **CURRENT ROW**: 현재 행
4. **n PRECEDING**: 현재 행을 기준으로 n행 이전
5. **n FOLLOWING**: 현재 행을 기준으로 n행 이후

5. 고급 활용 예제

5.1 누적 집계

```
-- 월별 누적 매출
SELECT 년월, 매출액,
    SUM(매출액) OVER (
        ORDER BY 년월
        ROWS UNBOUNDED PRECEDING
    ) as 누적매출
FROM 월별매출;
```

5.2 이동 평균

```
-- 3개월 이동평균
SELECT 년월, 매출액,
    AVG(매출액) OVER (
        ORDER BY 년월
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
    ) as 이동평균
FROM 월별매출;
```

5.3 그룹 내 비율

```
-- 부서별 급여 비율
SELECT 부서명, 사원명, 급여,
    RATIO_TO_REPORT(급여) OVER (PARTITION BY 부서명) * 100 as 비율
FROM 사원;
```

6. 성능 고려사항

6.1 최적화 포인트

1. **적절한 인덱스 사용**
   - ORDER BY 절에 사용되는 컬럼의 인덱스
2. **파티션 크기 고려**
   - 너무 큰 파티션은 성능 저하의 원인
3. **윈도우 범위 제한**
   - 필요한 범위만큼만 지정

SQLD 시험 대비 TIP

주요 출제 포인트

1. **순위 함수의 차이점**
   - ROW_NUMBER, RANK, DENSE_RANK의 결과 차이
2. **윈도우절의 이해**
   - ROWS와 RANGE의 차이
   - 각종 윈도우 지정 옵션의 의미
3. **복합적인 활용**
   - 다중 파티션
   - 복잡한 윈도우 범위 설정

학습 전략

1. 각 함수의 특징과 차이점 이해
2. 다양한 예제로 실습
3. 결과 집합 예측 연습
4. 성능 관련 내용 숙지

실전 문제 유형

1. 순위 함수 결과 비교
2. 누적 집계 값 계산
3. 이동 평균 구하기
4. 비율 계산