**쿼리 변환**

1. **쿼리 변환이란?**

- 휴리스틱 쿼리 변환 : 결과만 보장된다면 무조건 쿼리 변환 수행. 일종의 RBO. 경험적으로 항상 더 나은 성능을 보일 것이라는 옵티마이저 개발팀의 판단 반영

- 비용 기반 쿼리 변환 : 변환된 쿼리의 비용이 더 낮을 때 사용하며 낮이 않을 경우 원본 쿼리를 이용해 최적화 한다.

**2. 서브쿼리 UNNESTING**

- NESSTED 서브쿼리는 메인 쿼리와 부모 자식이라는 종속적이고 계층적인 관계이며 논리적인 관점에서 그 처리과정은 IN, EXISTS를 불문하고 NL 조인과 비슷한 필터 방식을 이용하며 이 방식이 항상 최적의 수행 속도를 보장 못함

- 옵티마이저는 동일한 결과를 보장하는 조인문으로 변환 후 최적화하는 서브쿼리 UNNESTING 혹은 원본 쿼리를 이용해 최적화하여 서브쿼리에 필터 오퍼레이션 발생.

- Unnesting의 이점 : 옵티마이저는 많은 조인 테크닉을 보유하게 되어 더 나은 실행계획을 찾을 가능성이 증가한다.

- Unnest Hint : UNNEST 혹은 NO\_UNNEST를 힌트에 기술하여 사용여부 결정.

- Unnest가 항상 유리한 것은 아니며 옵티마이저가 COST 산정 후 결정한다.

- 서브쿼리가 M쪽 집합일 경우 서브쿼리 UNNESTING결과가 서브쿼리 수준으로 (M레벨) 출력되어 이 경우 메인쿼리가 먼저 드라이빙 될 경우 세미조인 방식을 사용하며 서브쿼리가 그럴 경우 SORT UNIQUE 오퍼레이션으로 1쪽 집합 만든 후 조인.

**3. 뷰 Merging**

- 두 개 이상의 인라인 뷰가 있을 때 하나의 쿼리로 만들어 실행하는 것. 이 경우 옵티마이저는 더 다양한 액세스 경로를 조사할 수 있다.

- HINT로는 MERGE, NO\_MERGE를 힌트로 기술해주면 된다.

- 단순한 뷰는 Merging해도 성능이 나빠지지 않는다.

- 복잡한 연산을 포함하는 뷰 Merging은 성능이 나빠질 수 있다 (GROUP BY. DISTINCT)

- 쿼리 수행 비용 조사 후 적용 여부를 판단하는 방식으로 옵티마이저가 발전 중

- 뷰 Merging이 불가능 한 경우

1) 집합 연산자 2) CONNECT BY 절 3) ROWNUM 수도 컬럼 4) 집계함수 5) 분석함수

**4. 조건절 PUSHING**

- 옵티마이저는 1차적으로 뷰 Merging를 수행하지만 상황에 따라 조건절 PUSHING 시도.

- 조건절 PUSHING 기술

1) 조건절 PUSH DOWN : MAIN QUERY의 조건을 QUERY블락으로 밀어 넣는 것

2) 조건절 PULL UP : QUERY BLOCK안의 조건을 MAIN QUERY로 꺼내 오는 것. 혹은 다시 다른 QUERY BLOCK안으로 PUSHDOWN 하는데 활용

3) 조인조건 PUSHDOWN : NL 조인 수행 중 OUTER의 조건을 INNER쪽으로 뷰 쿼리 안으로 밀어 넣는 것.

- 옵티마이저 힌트로는 PUSH\_PRED / NO\_PUSH\_PRED를 사용한다.

**5. 조건절 이행**

E.DEPNO = D.DEPNO AND E.DEPNO = 10이라면 D.DEPNO = 10조건을 추가하여 해쉬 조인 또는 소트 머지 조인을 수행하기 전에 필터링 되어 조인 량이 감소하며 D 테이블 엑세스를 위한 인덱스 사용 추가 검토도 가능해진다.

**6. 불필요한 조인 제거**

- 1:M 관계에서 1쪽 집합을 읽지 않고 두 테이블간 PK – FK 제약조건이 존재하며 NOT NULL제약조건이라면 1쪽 집합 조인이 불필요하다.

- 위와 같은 이유 때문에 징검다리 튜닝을 할 경우 힌트를 주어 실행하지 못하게 해야한다.

**7. OR조건을 UNION으로 변환**

- 각각의 인덱스를 활용할 수 있게 해줌

**8. 기타 쿼리 변환**

1) 집합 연산을 조인으로 변환 : MINUS를 NOT EXISTS, INTERSECT를 EXISTS로.

2) 조인 칼럼에 IN NOT NULL 추가 : COUNT시 NULL포함 안되므로 NOT NULL 사용을 통해 결과집합에서 제외 후 COUNT

3) 필터조건 추가 : MAX < MIN 인 비정상적 조건인 경우 실제 실행결과 읽은 블락은 0.

4) 조건절 비교 순서 : 최신 DBMS는 일량을 비교하여 선행 처리 조건 결정.

- 힌트로는 QUERY\_TRANSFORMATION / NO\_QUERY\_TRANSFORMATION을 사용