**SQL 파싱 부하**

**1. 사용자가 SQL을 실행시 SQL PARSE**

- SYNTAX, SEMANTIC검사, LIBRARY CACHE에 실행계획 있는지 점검

- SOFT / HARD PARSING : SOFT는 LIBRARY CACHE에 찾아서 실행. HARD는 최적화 과정을 거치고 실행. (실행계획을 찾지 못했으므로) -> 최적화 과정은 엄청난 비용이 듦.

**2. 캐싱된 SQL 공유**

1) 실행계획 공유 조건

- 문법, 의미 오류 없는지 찾은 후 해쉬 함수로부터 반환된 해쉬 값으로 LIBRARY CACHE 내 HASH BUCKET 검색 후 HASH CAHIN 엔트리를 차례로 SCAN. 찾으면 저장된 계획으로 실행하고 없으면 최적화 수행 후 방금 탐색한 버켓의 체인에 연결 후 SQL 실행

2) 실행계획을 공유하지 못하는 경우

- 공백 또는 줄 바꿈. 대 소문자 구분, 주석, 테이블 OWNER 표시. 조건절 비교값 (BIND변수쓰면 해결!)

**3. 바인드 변수 사용하기**

- 오라클의 경우 바인드 변수를 명시적 사용 필요. SOFT PARSING을 하기 때문에 HARD PARSING은 1회만 일어남. SQL SERVER의 경우 2천번까지 자동 활성화

- 바인드 변수를 사용하지 않아도 되는 경우 : 정보계(DW,OLAP) 시스템에서 사용하는 LONG QUERY, DISTINCT VALUE가 소수일 경우

**4. STATIC SQL과 DYNAMIC SQL**

- STATIC SQL : SQL을 STRING변수에 담아 DBMS에 전달하지 않음

- DYNAMIC SQL : SQL을 STRING 변수에 담아 DBMS에 전달

- LIBRARY CACHE의 성능을 논할 땐 둘 차이 X. 바인드 변수 사용여부가 중요!

**5. APPLICATION CURSOR CACHING**

- C언어의 경우 HOLD\_CURSOR = YES 와 RELEASE\_CURSOR = NO 사용 후 YES로 바꿈

- 자바의 경우 PREPAREDSTATEMENT STMT 형식으로 사용한다.

- PL / SQL의 경우 자동적으로 실행되며 DYNAMIC SQL으론 안된다.