

## 1. Parse - 구문분석

Syntax Check 문법검사: 오라클 키워드 검사

Semantic Check 의미검사: 테이블이름이나 칼럼이름 검사

Shared Pool Check 권한검사: 해당 사용자가 이 테이블을 조회할 수 있는지 검사

검사들은 데이터 덱서너리를 사용하게 되고

데이터 덱서너리를 캐싱해 두고 성능을 높여주는 곳이 Shared Pool 안의 Dictionary Cache 또는 Row Cache

Shared Pool의 Library Cache를 검사한 후 공유되어 있는 실행계획이 있는지 검사

있으면 바로 Execution 단계로 진행 --> **Soft Parse** 또는 커서 공유라고 함

없으면 Optimizer를 찾아가서 실행 계획을 만들어 달라고 요청 --> **Hard Parse**

### Soft Parse

Library Cache안에 있는 커서 즉, 공유커서는

이미 한번 수행되었던 SQL 문장의 실행계획과 관련 정보를 보관 하고 있다가 재사용 해줌

Optimizer 옵티마이저가 실행계획을 만들어주는 부담을 줄이게 됨으로써 SQL 수행 속도를 빠르게 함

커서공유는 Parent Cursor 와 Child Cursor 모두가 완전히 동일해야만 이루어지게 됨

### Hard Parse

Optimizer 는 실행 계획을 생성시켜주는 역할

실행 계획을 세울 때 데이터 덱서너리(Static Data Dictionary) 정보를 보고 판단

옵티마이저가 실행 계획을 세워주는 대로 실행을 하기 때문에 옵티마이저의 실행 계획이 SQL 수행 속도에 절대적인 영향

Static Data Dictionary는 자동으로 업데이트 되지 않기 때문에 직접 관리 해줘야 함

인덱스 생성과 공간 관리 역시 사람이 관리 해줘야 함

## 2. Execute - 실행

하드 디스크의 데이터 파일에서 데이터가 들어있는 블록을 찾아 메모리 즉, DB Buffer Cache로 복사해 오는 과정

서버 프로세스가 해당 데이터를 가져오기 위해 해당 데이터가 저장되어 있는 블록을 찾게 됨

찾는 모든 데이터는 Database Buffer Cache에 있어야 함

서버 프로세스는 해당 블록을 찾기 위해서 우선 DB Buffer Cache를 확인하는데

① DB Buffer Cache에 원하는 블록이

있으면 즉시 다음단계인 Fetch 단계로 진행

② 없으면 서버 프로세스가 하드 디스크로 가서 해당 블록을 찾아 DB Buffer Cache로 복사  
디스크에서 메모리로 복사해 오는 시간이 굉장히 오래걸리는 편 -> 성능에 영향을 준다.

인덱스 생성 해놓게 되면 디스크에 어떤 파일이 어디있는지 알기 때문에 성능 UP, 속도 UP

옵티마이저가 실행계획을 세울 때 인덱스가 있으면 인덱스를 보고 찾아가라고 하기 때문에 인덱스 생성은 성능과 연관됨

인덱스가 없다면 풀스캔을 해야하기 때문에 성능 DOWN

## 3. Fetch - 인출

DB Buffer Cache에 복사된 블록 중에서 사용자가 요청한 원하는 데이터만 골라내는 과정