**MySQL 서버 = MySQL 엔진 + 스토리지 엔진**

MySQL 서버는 크게 MySQL 엔진과 스토리지 엔진으로 나눠볼 수 있다.

MySQL 엔진은 클라이언트로부터 오는 요청 처리(요청된 SQL 문장을 분석, 최적화, ...)를 담당하고, 스토리지 엔진은 실제 데이터를 디스크 스토리지에 저장하거나 조회하는 부분을 담당한다.

* **MySQL 엔진**
  + Connection Handler : 커넥션 및 쿼리 요청을 처리 담당
  + SQL 인터페이스 : DML, DDL, Procedure, View 등 SQL 인터페이스 제공 담당
  + SQL 파서(parser) : SQL문법 오류 탐지 및 SQL 쿼리 문장을 MySQL이 처리하기 좋은 토큰 단위로 나눠서 트리 형태로 파싱 하는 작업 담당
  + SQL 옵티마이저(optimizer) : 쿼리의 최적화된 실행 담당
  + 캐시와 버퍼 : 성능 향상을 위한 보조 저장소 기능 담당
* **스토리지 엔진**
  + 스토리지 엔진은 MySQL 엔진과 플러그인 형태로 연동/분리 가능하고 핸들러 API(핸들러 요청)를 통해 스토리지 엔진에 읽기/쓰기 요청이 가능

**MySQL 스레드 구조**

MySQL 스레드 = foreground thread + background thread

* **포그라운드 스레드**
  + 클라이언트/사용자가 요청한 쿼리 문장을 처리하는 스레드(고객 창구 역할)
  + 데이터 조회 스레드(데이터 버퍼나 캐시 또는 직접 디스크나 인덱스 파일로 접근하여 데이터를 가져오는 스레드)
* **백그라운드 스레드**
  + Insert buffer를 병합하는 스레드
  + 로그 기록 스레드
  + InnoDB 버퍼 풀의 데이터를 디스크로 기록(Write)하는 스레드
  + 데이터를 버퍼로 읽어들이는 스레드
  + 잠금, 데드락을 모니터링하는 스레드
  + 모든 백그라운드 스레드를 관리하는 메인 스레드

포그라운드 스레드는 'thread\_cache\_size' 설정 값에 따라 일정 스레드 개수로 유지된다.

**메모리 구조와 할당**

MySQL의 메모리 구조는 크게 글로벌 메모리 영역과 로컬(세션) 메모리 영역으로 나눠볼 수 있다.

글로벌 영역은 많은 스레드에 의해 공유되는 영역이고 로컬 영역은 많은 스레드에 의해 공유되지 않는 독립적인 영역이다.

* 글로벌 메모리
  + MySQL 서버를 실행할 때 운영체제로부터 설정한 만큼 할당받는 영역
  + 모든 스레드로부터 공유되는 영역
* 로컬(세션) 메모리
  + 클라이언트 스레드가 쿼리를 처리하는데 쓰이는 영역
  + 스레드 별로 독립적이라 절대 공유되지 않는 영역
  + 쿼리의 용도 별로 할당이 될 때도 있고 아예 할당되지 않을 때도 있음

**쿼리 실행 구조**

* 파서(Parser)가 들어온 쿼리 문장에 대해서 **문법(Syntax) 확인**을 하고 쿼리 문장을 적절한 단위로 분할하여 **파서 트리**(parser tree)를 만든다.
* 전처리기(pre-processor)가 앞에서 만들어진 **파서 트리**를 보고 **구조적으로 문제가 있는지 확인**한다. (예를 들면 없는 테이블, 컬럼 등에 접근한다거나 테이블에 접근하려고 보니 권한이 없다거나 하는 등...)
* 옵티마이저(Optimizer)가 요청받은 쿼리를 어떻게하면 **가장 적은 비용**으로 **빠르게** 즉, 최적으로 처리할지를 결정한다. 앞으로 개발자가 해야 할 일은 이 옵티마이저가 더 나은 결정(실행 계획 생성)을 할 수 있도록 쿼리를 작성하는 것이다.
* 실행 엔진은 옵티마이저에 의해 결정된 **실행 계획**대로 핸들러에게 지시한다. (임시테이블 만들어라 → where 조건대로 레코드 읽어와라 → 읽은 레코드를 임시 테이블에 써라 → ...)
* 핸들러(스토리지 엔진)는 실행 엔진이 내려준 지시대로 작업을 수행한다.