이번 토픽에서는 데이터베이스에 저장된 데이터를 분석하기 위해 SELECT로 시작하는 SQL 문을 어떻게 사용해야하는지 자세히 배워봤습니다.

(데이터 조회/분석 이외에 데이터 삽입, 갱신, 삭제 등을 위한 SQL 문은 다음 토픽에서 배웁니다, 데이터 조회/분석 뿐만 아니라 해당 기능까지도 필요하신 분들은 곧 출시될 다음 토픽을 기다려주세요)

그런데 여러분이 어떤 회사에 데이터 분석 직군으로 입사했다고 해도 바로 SELECT 문을 사용할 수 있는 건 아닙니다. 일단 이 회사의 데이터가 어떻게 관리되고 있는지부터 파악을 해야하기 때문이죠.

회사의 데이터 저장 상태를 파악할 때는 기존 직원 분들의 설명을 듣고, 문서화된 자료를 읽는 것이 가장 좋습니다. 그리고 그것과 동시에 데이터베이스 현황을 간단하게 파악할 수 있는 SQL 문을 알고 직접 적용해보는 게 좋은데요.

데이터베이스의 현황을 파악하려면 일단 기본적으로

회사의 서버에

**(1) 어떤 데이터베이스들이 있는지**

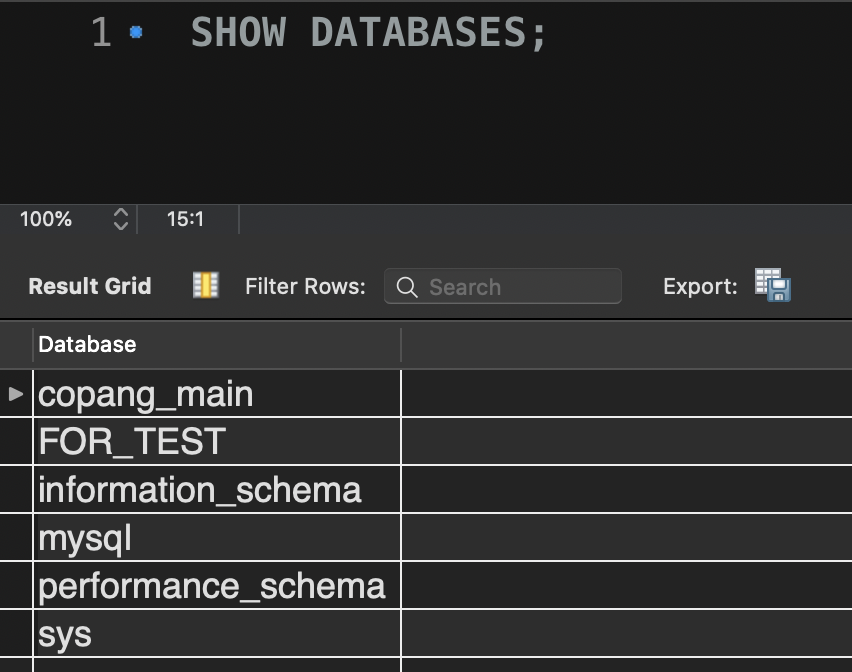
**(2) 각 데이터베이스 안에 어떤 테이블들이 있는지**

**(3) 각 테이블의 컬럼 구조는 어떻게 되는지**

**(4) 테이블들 간의 Foreign Key 관계는 어떤지**

등을 조사해야합니다. DBMS로 MySQL을 사용하는 곳이라면 각 작업을 어떻게 할 수 있을지 간단히 보여드릴게요.

**1. 존재하는 데이터베이스들 파악**

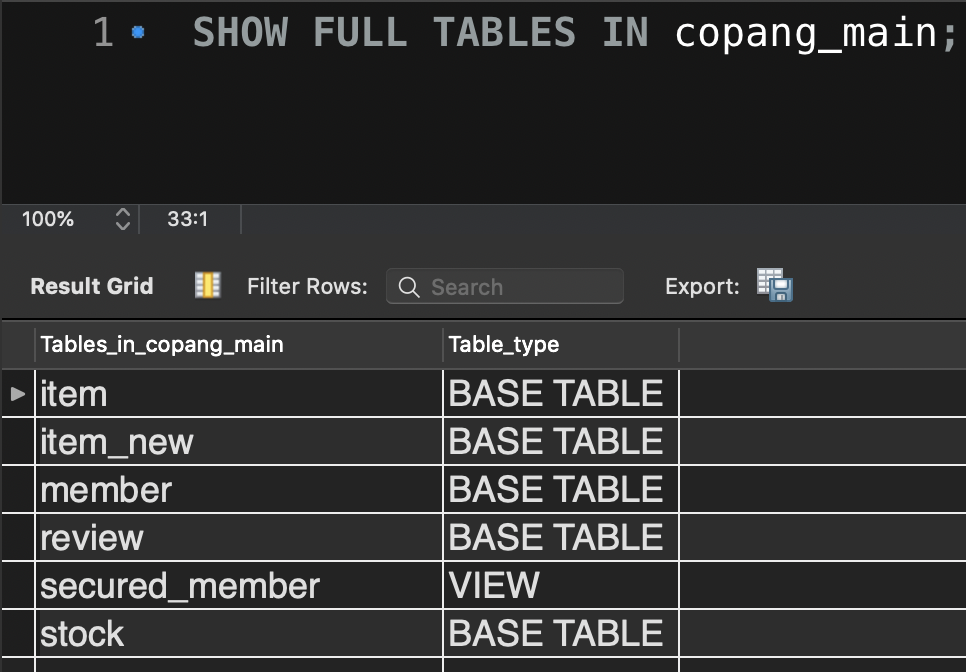


일단 현재 DBMS 상에 존재하는 데이터베이스들을 파악해야 합니다. 지금 위 결과에서는

**copang\_main** / **FOR\_TEST** 이것들이 사용자가 만든 데이터베이스이고,

나머지 **information\_schema** / **mysql** / **performance\_schema** / **sys**가 MySQL이라는 DBMS의 구동을 위해 원래부터 존재하는 기본 데이터베이스들입니다. 개발자 분들의 경우, 이 기본 데이터베이스들을 참조하시면 DBMS에 관해 깊은 공부를 하실 수 있을 겁니다.

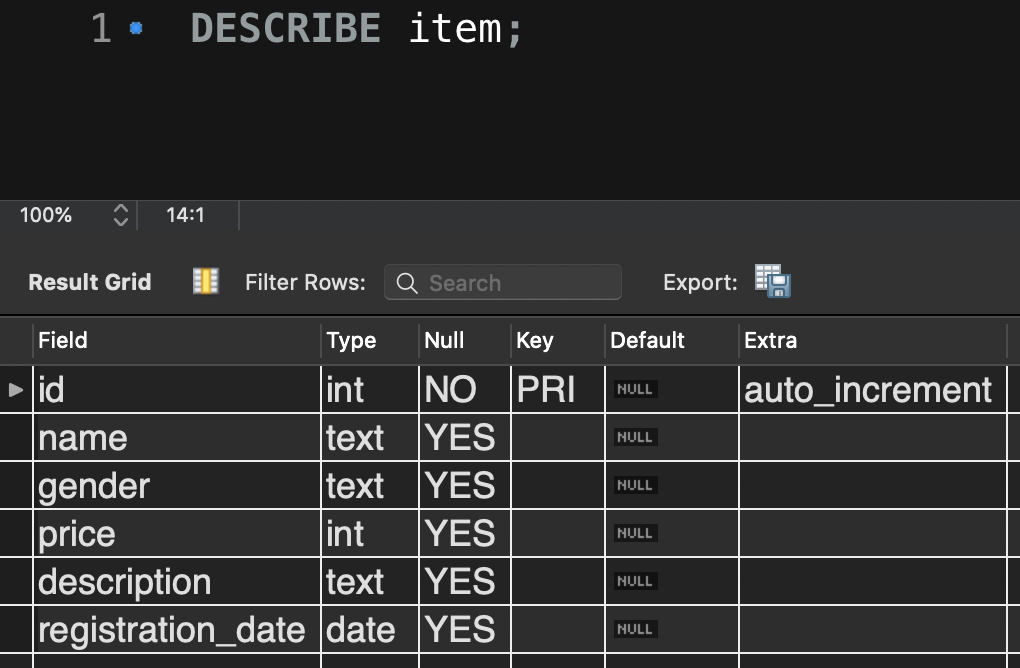
**2. 한 데이터베이스 안의 테이블(뷰도 포함)들 파악**



한 데이터베이스 안에 어떤 테이블, 어떤 뷰들이 있는지 파악하는 것도 중요합니다. 지금 보면 테이블들은 **BASE TABLE**이라고 표시되어 있고, 뷰는 **VIEW**라고 표시되어 있습니다. 이전에도 말했다시피 실무에서 여러분은 테이블에 직접 접근할 수는 없고, 뷰만 받게 되실 수도 있습니다.

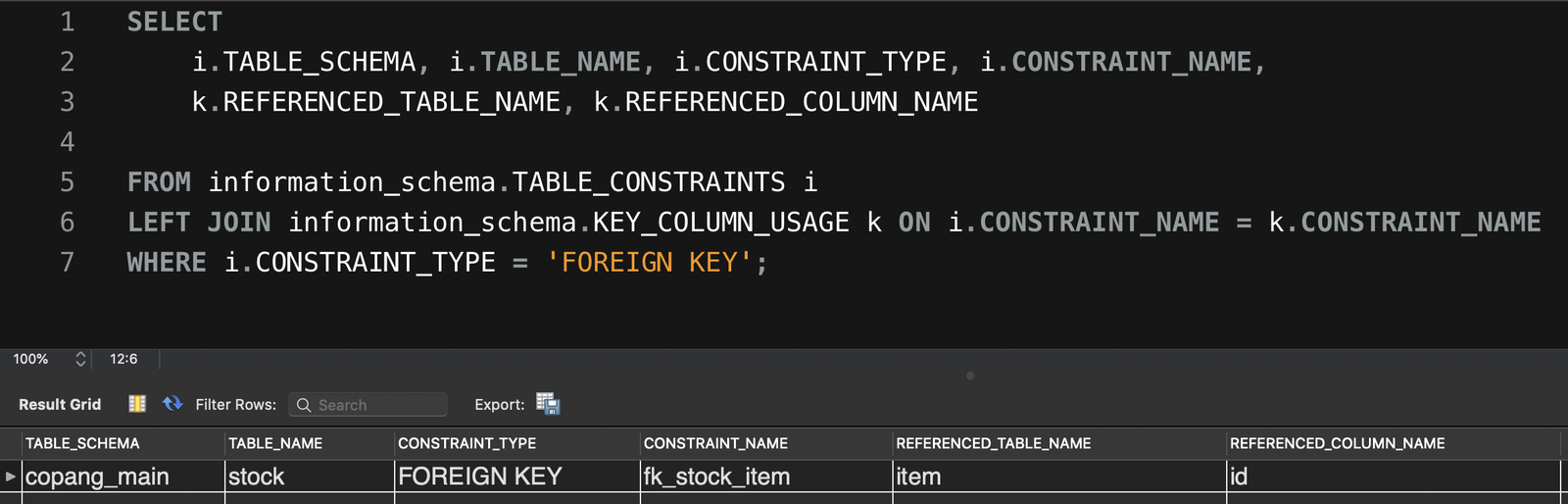
**3. 한 테이블의 컬럼 구조 파악**

한 테이블을 살펴보려면 **SELECT \* FROM 테이블 이름;** 을 실행해도 되겠지만 간단하게 컬럼 구조만 살펴볼 수 있게 해주는 SQL 문도 있습니다.



지금 **item** 이라는 테이블의 컬럼 구조를 파악하기 위해 **DESCRIBE**라는 키워드를 사용했습니다. 각 컬럼의 이름(Field)과 데이터 타입(Type), Not Null 속성 유무(Null), Primary Key 여부(Key) 등이 표시되어 있습니다. **DESCRIBE**를 사용하면 테이블의 컬럼 구조만 깔끔하게 파악할 수 있어서 좋습니다.

**4. Foreign Key(외래키) 파악**



테이블들 간의 관계를 파악하려면 데이터베이스에 존재하는 **Foreign Key**들을 파악해야합니다. 위 SQL 문은 MySQL가 직접 관리하는 기본 데이터베이스에서 Foreign Key 관련 정보를 꺼내오는 SQL 문인데요. Foreign Key 관련 정보를 조회하는 SQL 문은 DBMS의 기본 데이터베이스에서 그 정보를 가져오는 것이기 때문에 DBMS마다 그 차이가 큽니다. 본인이 사용하는 DBMS에 맞는 SQL 문을 스스로 검색해보세요.

그런데 Foreign Key를 파악할 때는 한 가지 문제가 있습니다. 그건 바로 두 테이블의 각 컬럼 간에 **Foreign Key 관계가 성립한다고 해도 관리자가 그것을 Foreign Key로 설정하지 않는 경우도 많다**는 건데요. 관리자의 실수때문에 그런 것일 수도 있고, 데이터베이스의 성능을 고려해서 의도적으로 그렇게 하는 경우도 있습니다. 개발자가 아닌 분에게는 너무 어려운 내용이기 때문에 설명은 생략하겠습니다. 일단, **Foreign Key 관계가 논리적으로 성립해도 실제로 DBMS 상에서 설정되어 있지 않은 경우도 많다는 걸 기억하세요.**

따라서 Foreign Key들을 정확하게 파악하려면, 해당 회사의 데이터베이스를 설계한 분의 설명을 듣거나, 본인이 직접 데이터의 관계 및 흐름을 파악해서 스스로 파악할 수밖에 없습니다.

자, 이때까지 데이터베이스 현황을 간단하게 파악하는 방법들을 배웠습니다. 그런데 이런 방법들보다도 훨씬 효율적인 방법이 하나 더 있습니다. 그건 바로 그 회사에서 이미 사용되고 있는 기존의 SQL 문들을 자세하게 살펴보는 것입니다. 그럼 이 회사에서 필요로 하는 데이터의 성격은 어떤 것인지, 필요한 데이터들은 주로 어느 테이블에 있는지 등과 같은 정보를 빠르게 파악할 수 있을 겁니다.

데이터베이스 현황 파악이 어느 정도 끝나면, 이번 토픽에서 배운 SELECT 문으로 본격적인 실력 발휘를 하시면 되겠죠?