

7.1 쿼리와 연관된 시스템 설정

🕒 생성일	@2021년 7월 3일 오후 5:41
⋮ 태그	

쿼리 성능 향상을 위해 쿼리 패턴별로 어떻게 처리하는지 이해하고, 유용한 패턴을 찾아보자

쿼리와 연관된 시스템 설정

MySQL 서버의 시스템 설정에 따라 SQL 작성 규칙이 달라진다

1. SQL 모드

여러 개의 값이 동시에 설정될 수 있음

STRICT_ALL_TABLES

일반적으로 MySQL에서는 저장하려는 값의 길이가 칼럼의 길이보다 더 긴 경우라 하더라도 에러가 발생하지 않는다. 칼럼의 길이를 초과하는 부분은 버리고 저장 가능한 만큼만 칼럼에 저장한다. 물론 경고 메시지가 발생하지만 이를 관심 있게 보는 사용자는 많지 않을 것이다. 가끔 이것이 문제가 되기도 하는데, `sql_mode` 시스템 변수에 `STRICT_ALL_TABLES`가 설정되면 칼럼의 정해진 길이보다 큰 값을 저장할 때 경고가 아닌 오류가 발생하고 쿼리 실행이 중지된다.

STRICT_TRANS_TABLES

칼럼의 타입과 호환되지 않는 값을 저장할 때, MySQL 서버는 비슷한 값으로 최대한 바꿔서 저장하려고 한다. 하지만 `STRICT_ALL_TABLES`와 비슷하게 이러한 부분이 사용자를 오히려 더 혼란스럽게 하는 원인이 되기도 한다. `STRICT_TRANS_TABLES`를 설정하면 원하지 않는 데이터 타입의 변환이 필요할 때 MySQL 서버는 강제 변환하지 않고 에러를 발생시킨다.

TRADITIONAL

`STRICT_TRANS_TABLES`나 `STRICT_ALL_TABLES`와 비슷하지만 조금 더 엄격한 방법으로 SQL의 작동을 제어한다. `STRICT_ALL_TABLES`, `STRICT_TRANS_TABLES`, `TRADITIONAL` 등의 설정은 MySQL 서버가 조금 더 ANSI 표준 모드로 작동하도록 유도한다.

ANSI_QUOTES

MySQL에서는 문자열 값(리터럴)을 표현하기 위해 홑따옴표와 쌍따옴표를 동시에 사용할 수 있다. 하지만 오라클과 같은 DBMS에서는 홑따옴표는 문자열 값을 표기하는 데 사용하고, 쌍따옴표는 칼럼 명이나 테이블 명과 같은 식별자(Identifier)를 구분하는 용도로만 사용한다. 이 또한 MySQL에 익숙하지 않은 사용자에게는 혼란스러울 수 있다. 때로는 MySQL에 익숙하다 하더라도 하나의 SQL 문장에서 홑따옴표와 쌍따옴표가 엉켜 있으면 가독성이 떨어지기도 한다.

`sql_mode` 시스템 설정에 `ANSI_QUOTES`를 설정하면 홑따옴표만 문자열 값 표기로 사용할 수 있고, 쌍따옴표는 칼럼명이나 테이블명과 같은 식별자(Identifier)를 표기하는 데만 사용할 수 있다. 더 자세한 내용은 7.3.1절 "리터럴 표기법"(382쪽)을 참조하자.

ONLY_FULL_GROUP_BY

MySQL의 쿼리에서는 `GROUP BY` 절에 포함되지 않은 칼럼이라도 집합 함수의 사용 없이 그대로 `SELECT` 절이나 `HAVING` 절에 사용할 수 있다. 이러한 부분도 SQL 표준이나 다른 DBMS와는 다른 동작 방식인데, `sql_`

PIPE_AS_CONCAT

MySQL에서 "||"는 OR 연산자와 같은 의미로 사용된다. 하지만 sql_mode 시스템 설정에 PIPE_AS_CONCAT 값을 설정하면 오라클과 같이 문자열 연결(CONCAT) 연산자로 사용할 수 있다.

PAD_CHAR_TO_FULL_LENGTH

MySQL에서는 CHAR 타입이라 하더라도 VARCHAR와 같이 유효 문자열 뒤의 공백 문자는 제거되어 반환된다. 이는 주로 애플리케이션 개발자에게 민감한 부분인데, 개인적으로 저자는 MySQL이 불필요한 공백 문자를 제거하는 방식이 더 편리한 것 같다. 하지만 CHAR 타입의 칼럼값을 가져오는 경우, 뒤쪽의 공백이 제거되지 않고 반환돼야 한다면 sql_mode 시스템 설정에 PAD_CHAR_TO_FULL_LENGTH를 추가하면 된다. 더 자세한 내용은 15.1절, "문자열(CHAR와 VARCHAR)"(866쪽)에서 언급하겠다.

NO_BACKSLASH_ESCAPES

MySQL에서도 일반적인 프로그래밍 언어에서처럼 역 슬래시 문자를 이스케이프 문자로 사용할 수 있다. sql_mode 시스템 설정에 NO_BACKSLASH_ESCAPES를 추가하면 역 슬래시를 문자의 이스케이프 용도로 사용하지 못한다. 이 설정을 활성화하면 백 슬래시 문자도 다른 문자와 동일하게 취급한다. 더 자세한 내용은 15.1절, "문자열(CHAR와 VARCHAR)"(866쪽)에서 언급하겠다.

IGNORE_SPACE

MySQL에서 스토어드 프로시저나 함수의 이름 뒤에 공백이 있으면 "스토어드 프로시저나 함수가 없습니다"라는 에러가 출력될 수도 있다. MySQL에서는 스토어드 프로시저나 함수명과 괄호 사이에 있는 공백까지도 스토어드 프로시저나 함수의 이름으로 간주한다. 이 동작 방식이 기본 모드이므로 몇 번이고 함수가 있는지 확인하기도 한다. sql_mode 시스템 설정에 IGNORE_SPACE를 추가하면 프로시저나 함수명과 괄호 사이의 공백은 무시한다.

ANSI

이 값은 위에서 설명한 여러 가지 옵션을 조합해서 MySQL 서버가 최대한 SQL 표준에 맞게 동작하게 만들어준다.

대강.. 이렇게 있다고만 생각하면 될듯하다

2. 영문 대소문자 구분

lower_case_table_names 시스템 변수값으로 대소문자 구분 유무를 설정할 수 있다. → 디폴트는 0 (대소문자 구분)

3. MySQL 예약어

테이블/칼럼 명이 예약어와 같은 키워드로 생성하면,, 역따옴표`나 쌍따옴표로 래핑해줘야 함