


16.5 테이블 파티션

🕒 생성일	@2021년 8월 28일 오후 6:31
☰ 태그	

- 5.1, 5.5 버전의 파티션 기능은 상당히 제약이 많음
- 파티션 기준 값이 날짜, 숫자타입이면 ⇒ 레인지 파티션, 리스트 파티션을 사용하면 됨

만약 파티션의 기준 값이 날짜나 숫자 이외의 타입이라면?

예를들어 user_id같은 CHAR타입이면,,? ⇒ 키() 파티션을 사용하자

```
PRIMARY KEY (user_id, action_type, action_dttm),  
INDEX ix_userid_targetobjectid (user_id, target_object_id)  
) ENGINE=INNODB  
PARTITION BY KEY() PARTITIONS 20 ;
```

이럴 경우.. `PARTITION BY KEY()` 처럼 괄호가 비어있으면 프라이머리 키의 모든 칼럼이 파티션 칼럼이 파티션 키가 됨

- 하지만 위 경우에는 파티션 프루닝 기능을 이용할 수 없다

```
partitions  
SELECT * FROM tb_user_action_history  
WHERE user_id='toto';
```

위 쿼리의 실행 계획을 확인해 보면 다음과 같이 partitions 칼럼에 모든 파티션의 목록이 나열되면서 파티션 프루닝이 적용되지 않고 모든 파티션이 검색 대상이 되어 버린다는 것을 알 수 있다.

id	select_type	Table	partitions	...	Extra
1	SIMPLE	tb_user_action_history	p0,p1,p2,p3,p4,p5,p6,p7, p8,p9,p10,p11,p12,p13, p14,...	...	Using where

위 경우에는 `PARTITION BY KEY()` 안에 user_id를 명시 해주면, 됨

