

자기주도 PJT

MySQL Event Scheduler 를 이용한 Scheduler 개발

목차

1. 과제 개요.....	3
2. 기본 과제.....	4
3. 심화 학습.....	7
4. 과제 제출.....	8

1. 과제 개요

프로젝트를 수행하다 보면 주기적으로(매일 0 시, 매시 정각 등), 또는 특정 일시(특정 시점으로부터 1 시간 후, 2023.12.31.23:59:59 등)에 데이터베이스에 작업이 발생해야하는 경우가 있습니다.

Server, OS 등에서도 Scheduler 사용이 가능하겠지만, MySQL 에서도 Event 를 통해 이러한 작업을 할 수 있는데, 이것을 Event Scheduler 라고 합니다.

이번 과제에서는 Evnet Scheduler 를 이용하여 원하는 상황에 작업이 실행되도록 개발해 봅니다.

2. 기본 과제

본 명세서의 기본과제는

1. Event Scheduler 생성 전 확인 사항
2. Event Scheduler 구문
3. 과제 안내

로 구성되어 있습니다.

1. Event Scheduler 생성 전 확인 사항

- 1) MYSQL 서버의 Event Scheduler 사용여부 확인

```
SHOW VARIABLES LIKE 'event%';
```

- 2) value 가 OFF 면 ON 으로 변경

```
SET GLOBAL event_scheduler = ON;
```

- 3) MYSQL 내에 저장된 Event Scheduler 가 있는지 확인

```
SHOW EVENTS;
```

```
SELECT * FROM information_schema.EVENTS;
```

2. Event Scheduler 구문

1) 공식 DOCUMENTATION 참고

<https://dev.mysql.com/doc/refman/8.0/en/create-event.html>

2) CREATE EVENT Statement

```
CREATE
    [DEFINER = user]
    EVENT
    [IF NOT EXISTS]
    event_name
    ON SCHEDULE schedule
    [ON COMPLETION [NOT] PRESERVE]
    [ENABLE | DISABLE | DISABLE ON SLAVE]
    [COMMENT 'string']
    DO event_body;
```

- [] 부분은 생략 가능. - 자세한 내용은 공식문서 참고
- event_name : 이벤트 이름
- schedule : 수행, 반복할 시간 및 기간

```
schedule: {
    AT timestamp [+ INTERVAL interval] ...
    | EVERY interval
    [STARTS timestamp [+ INTERVAL interval] ...]
    [ENDS timestamp [+ INTERVAL interval] ...]
}
```

AT : 수행할 시간

EVERY : 반복할 시간

STARTS, ENDS : 반복할 기간

Interval 종류

```
interval:
quantity {YEAR | QUARTER | MONTH | DAY | HOUR | MINUTE |
WEEK | SECOND | YEAR_MONTH | DAY_HOUR | DAY_MINUTE |
DAY_SECOND | HOUR_MINUTE | HOUR_SECOND | MINUTE_SECOND}
```

- event_body : 실제 수행될 Query 문

3) 과제 안내

- TABLE 생성 – 간단하게 Insert 가 가능한 Table 생성
- 1 분마다 데이터가 등록되는 Event Scheduler 작성
- 현재 시각으로부터 5 분 후 모든 데이터를 삭제하는 Event Scheduler 작성
- 특정 기간 동안만 반복 실행되는 Event Scheduler 작성
- Scheduler 가 정상적으로 작동하는지 Data 의 변화를 확인

3. 심화 학습

1. 선택 학습 중 '스케줄러 활용' 과제도 함께 공부하여 다양한 방법의 Scheduler 를 학습해 봅니다.
2. 과제에서는 단순하게 1 개 Query 문으로 해결이 되게 만들었지만, 실제로는 훨씬 복잡한 작업을 수행하게 되는 경우가 많습니다. 이런 경우 CREATE 구문에 여러 개의 Query 문 사용을 위해서는 반드시 delimiter 를 정의해 주어야 합니다. delimiter 를 사용할 경우의 문법에 대해서도 공부해 보도록 합니다.(delimiter 는 Event 뿐 아니라 Function, Stored Procedure 등에서 다양하게 사용되므로 개념을 알아두시길 바랍니다.)

4. 산출물 제출

1. <https://lab.ssafy.com/s10-study/self-project/> 의 "산출물 제출 가이드".docx 참조
2. 제출할 내역
 - 작성한 Table CREATE 문
 - 작성한 3 개 Event Scheduler CREATE 문
 - `SELECT * FROM information_schema.EVENTS;` 결과 캡처