

# 实习二 数据库约束设计

成员：杨仕博 张钧天 马千里 张成谦

```
%load_ext sql
```

```
import pymysql
pymysql.install_as_MySQLdb()
%sql mysql://stu2100013111:stu2100013111@162.105.146.37:43306
```

```
%sql use stu2100013111;
```

## 练习二 触发器设计

滑动窗口是流数据处理里面的基本单元，通常用于维护最近N个数据的聚合信息。可以把滑动窗口看成是一个定长队列，窗口中每新到一个数据，则会将最老的数据元素从窗口中顶出去。

我们设计触发器来计算滑动窗口的两个聚集函数：sum和max。

我们建立一个用来存放原始数据的originData(id, value)表，再建立两个存放最近N个数据的表，sum\_slidingWin(id, value)表和max\_slidingWin(id, value)表，分别用于计算滑动窗口的sum和max值。计算出来的聚集值存放在aggResult(sumRes, maxRes)中，初始时存在一行(0,0)。

以下是建表代码：

```
%%sql

SET @@foreign_key_checks=0;
-- 1. 初始化originData表

DROP TABLE IF EXISTS originData;

CREATE TABLE originData
(
    id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    value BIGINT NOT NULL
);

SET @@foreign_key_checks=1;

-- 2. 初始化max_slidingWin表
SET @@foreign_key_checks=0;
DROP TABLE IF EXISTS max_slidingWin;

CREATE TABLE max_slidingWin
```

```

(
    id BIGINT NOT NULL PRIMARY KEY,
    value BIGINT NOT NULL,
    constraint max_fk_max_sliding foreign key(id) references originData(id)
);

SET @@foreign_key_checks=1;

-- 3. 初始化sum_slidingWin表
SET @@foreign_key_checks=0;
DROP TABLE IF EXISTS sum_slidingWin;

CREATE TABLE sum_slidingWin
(
    id BIGINT NOT NULL PRIMARY KEY,
    value BIGINT NOT NULL,
    constraint sum_fk_sum_sliding foreign key(id) references originData(id)
);

SET @@foreign_key_checks=1;

-- 4. 初始化aggResult表

SET @@foreign_key_checks=0;
DROP TABLE IF EXISTS aggResult;

CREATE TABLE aggResult
(
    id BIGINT NOT NULL PRIMARY KEY,
    sumRes BIGINT NOT NULL,
    maxRes BIGINT NOT NULL,
    constraint agg_fk_agg_sliding foreign key(id) references originData(id)
);

INSERT INTO aggResult(id,sumRes,maxRes) values (0,0,0);

SET @@foreign_key_checks=1;

```

题目要求构造一个模拟器，定时产生往originData表里插入数据，限定每次插入一行。id被设置为自动单调递增以代表数据元素到达窗口的顺序，value可以是限定在一定区间内的随机产生的整数。

我们需要生成随机数，然后构造一个按时调用上述存储过程的事件。

```

%%sql

-- 5. 构造一个定时往originData里增加数据的模拟器
-- 5(1). 首先我们构造一个向originData中存数据的存储过程

DROP PROCEDURE IF EXISTS insertOriginData;

CREATE PROCEDURE insertOriginData()
BEGIN

```

```

-- 生成随机数的最大值
DECLARE max_random_val INT;
SET max_random_val = 100;
INSERT INTO originData ( value ) VALUES(
    FLOOR(RAND() * max_random_val + 1)
);
END;

-- 5(2). 其次我们构造一个按时调用上述存储过程的事件

-- 两次事件间隔的秒数
SET @no_seconds_between_events = 1;

-- 事件执行的时间
SET @no_seconds_event_runs = 100;

DROP EVENT IF EXISTS insertOriginDataEvent;

CREATE EVENT insertOriginDataEvent
ON SCHEDULE EVERY @no_seconds_between_events SECOND
STARTS CURRENT_TIMESTAMP + interval 1 second
ENDS CURRENT_TIMESTAMP + interval @no_seconds_event_runs second
ON COMPLETION PRESERVE
DO
BEGIN
CALL insertOriginData();
END;

```

然后定义触发器，在originData表有数据插入时更新三个表，代码如下：

```

-- 6. 定义一个触发器在每次插入originData的时候更新
-- max_slidingWin表

DROP TRIGGER IF EXISTS insertMaxSlidingWin;

CREATE TRIGGER insertMaxSlidingWin
AFTER INSERT ON sum_slidingWin
FOR EACH ROW
BEGIN
    -- 设置N的大小
    DECLARE N INT;
    SET N = 10;
    DELETE FROM max_slidingWin WHERE
max_slidingWin.value <= NEW.value;
    DELETE FROM max_slidingWin WHERE
max_slidingWin.id <= NEW.id - N;
    INSERT INTO max_slidingWin(id, value)
VALUES(NEW.id, NEW.value);
END;

-- sum_slidingWin表

```

```

DROP TRIGGER IF EXISTS insertSumSlidingWin;

CREATE TRIGGER insertSumSlidingWin
AFTER INSERT ON originData
FOR EACH ROW
BEGIN
    -- 设置N的大小
    DECLARE N INT;
    SET N = 10;
    DELETE FROM sum_slidingWin WHERE
    sum_slidingWin.id <= NEW.id - N;
    INSERT INTO sum_slidingWin(id, value)
    VALUES(NEW.id, NEW.value );
END;

-- aggResult表

DROP TRIGGER IF EXISTS updateResult;

CREATE TRIGGER updateResult
AFTER INSERT ON max_slidingWin
FOR EACH ROW
BEGIN
    DECLARE ressum INT;
    DECLARE resmax INT;
    SELECT sum(value) INTO ressum FROM sum_slidingWin;
    SELECT value INTO resmax FROM max_slidingWin LIMIT 1;
    INSERT INTO aggResult(id,sumRes,maxRes) values (NEW.id,ressum,resmax);
END;

```

下面开始测试：

```
%sql select * from originData
```

id value

1 99

2 61

3 42

4 9

5 26

6 87

7 58

8 47

9 55

10 73

11 71

12 70

13 14  
14 42  
15 4  
16 81  
17 53  
18 23  
19 56

```
%sql select * from sum_slidingWin
```

id value

10 73  
11 71  
12 70  
13 14  
14 42  
15 4  
16 81  
17 53  
18 23  
19 56

```
%sql select * from max_slidingWin
```

id value

16 81  
19 56

```
%sql select * from aggResult
```

id sumRes maxRes

0 0 0  
1 99 99  
2 160 99  
3 202 99  
4 211 99  
5 237 99  
6 324 99  
7 382 99  
8 429 99  
9 484 99  
10 557 99

11 529 87

12 538 87

13 510 87

14 543 87

15 521 87

16 515 81

17 510 81

18 486 81

19 487 81