**练习五(二) 索引与视图的创建**

学号 姓名

三、存储过程与触发器

1、创建store表格，命令如下：

CREATE TABLE IF NOT EXISTS  store (  `id` int(11) NOT NULL  AUTO\_INCREMENT,

name varchar(20) NOT NULL,

count  int(11) NOT NULL DEFAULT  1,

PRIMARY KEY (id)

 ) ENGINE=InnoDB  DEFAULT CHARSET=utf8  AUTO\_INCREMENT=7;

INSERT INTO store (`id`, `name`, `count`) VALUES  (1, 'android', 15),

(2, 'iphone', 14),

(3, 'iphone', 20),

(4, 'android', 5),

(5, 'android', 13),

(6, 'iphone', 13);

2、编写存储过程StatisticStore，统计iphone的总库存是多少，并把总数输出到控制台。

delimiter //   --默认情况下，delimiter是分号; delimiter // 定义结束符为”//”

drop procedure if exists StatisticStore;   --如果存储过程StatisticStore已经存在就删除。

CREATE PROCEDURE StatisticStore()   --创建存储过程

BEGIN

    declare c int;

    declare n varchar(20);

    declare total int default 0;

    declare done int default false;   --声明c,n,total和done变量。其中done为结束标志变量

    declare cur cursor for select name,count from store where name = 'iphone';   --声明cur游标

    declare continue HANDLER for not found set done = true;   --指定游标循环结束时的返回值

    set total = 0;   --设置初始值

    open cur;   --打开游标

    fetch cur into n,c;   --游标指向当前数据

    while(not done) do   --判断游标的循环是否结束

        set total = total + c;   --获取一条数据时，将total值进行累加操作，这里可以做任意你想做的操作，

        fetch cur into n,c;     --推进游标

    end while;

    close cur;   --关闭游标

    select total;   --输出结果

END;

call StatisticStore();  // --调用存储过程

delimiter ;

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+-------+

| total |

+-------+

| 47 |

+-------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

3、查看jxgl数据库的存储过程。

show procedure status where db=’jxgl’;

结果如下：

-> ;

+------+----------------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+

| Db | Name | Type | Definer | Modified | Created | Security\_type | Comment | character\_set\_client | collation\_connection | Database Collation |

+------+----------------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+

| jxgl | StatisticStore | PROCEDURE | root@localhost | 2022-05-09 14:55:59 | 2022-05-09 14:55:59 | DEFINER | | utf8mb4 | utf8mb4\_0900\_ai\_ci | utf8\_general\_ci |

+------+----------------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+

1 row in set (0.02 sec)

4、触发器的使用

（1）创建学生表student\_info和学生数目统计表student\_count：

触发器是一种与表操作有关的数据库对象，当触发器所在表上出现指定事件时，将调用该对象，即表的操作事件触发表上的触发器的执行。

CREATE TABLE student\_info (stu\_no INT(11) NOT NULL AUTO\_INCREMENT,

stu\_name VARCHAR(255) DEFAULT NULL,

PRIMARY KEY (stu\_no)

);

CREATE TABLE student\_count (student\_count INT(11) DEFAULT 0);

INSERT INTO student\_count VALUES(0);

（2）补全如下触发器代码，在向学生表INSERT数据时，学生数增加，DELETE学生时，学生数减少。

CREATE TRIGGER trigger\_student\_count\_insert

AFTER INSERT ON student\_info FOR EACH ROW

UPDATE student\_count SET student\_count=student\_count+1

CREATE TRIGGER trigger\_student\_count\_delete

AFTER DELETE ON student\_info FOR EACH ROW

UPDATE student\_count SET student\_count=student\_count-1

执行：INSERT INTO student\_info VALUES(NULL,'张明'),(NULL,'李明'),(NULL,'王明');

（3）执行如下命令，写出结果并理解触发器功能：

SELECT \* FROM student\_info;

+--------+----------+

| stu\_no | stu\_name |

+--------+----------+

| 1 | 张明 |

| 2 | 李明 |

| 3 | 王明 |

+--------+----------+

3 rows in set (0.00 sec)

SELECT \* FROM student\_count;

+---------------+

| student\_count |

+---------------+

| 3 |

+---------------+

1 row in set (0.00 sec)

DELETE FROM student\_info WHERE stu\_name IN('张明','李明');

Query OK, 2 rows affected (0.01 sec)

SELECT \* FROM student\_info;

+--------+----------+

| stu\_no | stu\_name |

+--------+----------+

| 3 | 王明 |

+--------+----------+

1 row in set (0.00 sec)

SELECT \* FROM student\_count;

+---------------+

| student\_count |

+---------------+

| 1 |

+---------------+

1 row in set (0.00 sec)