

MySQL的CURD(基础)

笔记本: Java17&18班上课笔记

创建时间: 2021/3/8 9:40

更新时间: 2021/3/8 21:46

作者: 1262913815@qq.com

MySQL的CURD(基础)

1、回顾

登录: `mysql -u root -p111111`

```
mysql> use testjava17_18;  
Database changed
```

```
mysql> show tables;  
+-----+  
| Tables_in_testjava17_18 |  
+-----+  
| exam_result             |  
| stu_test                |  
| student                 |  
+-----+  
3 rows in set (0.01 sec)
```

关于库的操作:

-- 删除数据库

`drop database if exists test;`

-- 创建数据库

`create database test;`

-- 显示数据库

`show databases;`

关于表的操作：

-- 显示表

```
show tables;
```

-- 删除表

```
drop table if exists test0308;
```

-- 创建表

```
create table test0308(  
    id int,  
    name varchar(10)  
);
```

主要围绕表的CURD进行学习：

1、插入

-- 全列 + 单行

```
insert into test0308 values(1,'bit');
```

```
insert into test0308 values(2,'bit2');
```

-- 指定列+多行

```
insert into test0308(id,name) values(3,'bit3'),(4,'bit4');
```

2、查询

-- 全列查询

```
select * from test0308;
```

-- 查询指定列

```
select name from test0308;
```

- 今天的任务

1、去重 distinct

```
mysql> select * from exam_result;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
2	孙悟空	87.5	78.0	77.0
3	猪八戒	88.0	98.0	90.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
6	孙权	70.0	73.0	78.5
7	宋公明	75.0	65.0	30.0

```
7 rows in set (0.01 sec)
```

-- 查询所有人的数学成绩

```
mysql> select math from exam_result;
```

math
98.0
78.0
98.0
84.0
85.0
73.0
65.0

```
7 rows in set (0.00 sec)
```

-- 去重

```
mysql> select distinct math from exam_result;
```

math
98.0
78.0
84.0
85.0
73.0
65.0

```
+-----+
6 rows in set (0.00 sec)
```

疑问？？ 这样的SQL语句是否也可以去重？

```
mysql> select id,name, distinct math from
exam_result;
ERROR 1064 (42000): You have an error in your SQL
syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use
near 'distinct math from exam_result' at line 1
```

以上这条语句位置不对！！ 需要把去重放在第一位！！

```
mysql> select distinct math, id,name from
exam_result;
+-----+-----+-----+
| math | id  | name  |
+-----+-----+-----+
| 98.0 | 1   | 唐三藏 |
| 78.0 | 2   | 孙悟空 |
| 98.0 | 3   | 猪悟能 |
| 84.0 | 4   | 曹孟德 |
| 85.0 | 5   | 刘玄德 |
| 73.0 | 6   | 孙权   |
| 65.0 | 7   | 宋公明 |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

那么，去重的原理到底是什么？记住了，去重 针对的是，所有的字段！！ 只有当select distinct math, id,name from exam_result;中的三个字段同时重复的时候，才会去重！！

2、排序

-- 将查询结果 按照，根据数学成绩进行从低到高排序的排序

```
mysql> select * from exam_result order by math;
```

id	name	chinese	math	english
7	宋公明	75.0	65.0	30.0
6	孙权	70.0	73.0	78.5
2	孙悟空	87.5	78.0	77.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
1	唐三藏	67.0	98.0	56.0
3	猪悟能	88.0	98.0	90.0

```
7 rows in set (0.00 sec)
```

或者:

```
mysql> select * from exam_result order by math asc;
```

id	name	chinese	math	english
7	宋公明	75.0	65.0	30.0
6	孙权	70.0	73.0	78.5
2	孙悟空	87.5	78.0	77.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
1	唐三藏	67.0	98.0	56.0
3	猪悟能	88.0	98.0	90.0

```
7 rows in set (0.00 sec)
```

-- 将查询结果按照, 根据数学成绩进行从高到低排序的排序

```
mysql> select * from exam_result order by math desc;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
3	猪悟能	88.0	98.0	90.0
5	刘玄德	55.5	85.0	45.0

4	曹孟德	82.0	84.0	67.0
2	孙悟空	87.5	78.0	77.0
6	孙权	70.0	73.0	78.5
7	宋公明	75.0	65.0	30.0

7 rows in set (0.00 sec)

博哥：desc 这个词 见了2次了！！！ desc实际上就是一个关键字！！！！

```
mysql> desc exam_result;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	YES			
name	varchar(20)	YES			
chinese	decimal(3,1)	YES			
math	decimal(3,1)	YES			
english	decimal(3,1)	YES			

关键字不要瞎用！ 假设你要创建一个表，这个表名叫desc

```
-- ESC 下面那个键
drop table if exists `desc`;
create table `desc` (
  id int,
  name varchar(10)
);
```

对于要排序的字段为NULL的时候：

```
mysql> insert into exam_result values(8,'唐三藏',67,NULL,80);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from exam_result;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
2	孙悟空	87.5	78.0	77.0
3	猪悟能	88.0	98.0	90.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
6	孙权	70.0	73.0	78.5
7	宋公明	75.0	65.0	30.0
8	唐三藏	67.0	NULL	80.0

8 rows in set (0.00 sec)

排序查询:

```
mysql> select * from exam_result order by math asc;
```

id	name	chinese	math	english
8	唐三藏	67.0	NULL	80.0
7	宋公明	75.0	65.0	30.0
6	孙权	70.0	73.0	78.5
2	孙悟空	87.5	78.0	77.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
1	唐三藏	67.0	98.0	56.0
3	猪悟能	88.0	98.0	90.0

8 rows in set (0.00 sec)

```
mysql> select * from exam_result order by math desc;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
3	猪悟能	88.0	98.0	90.0
5	刘玄德	55.5	85.0	45.0
4	曹孟德	82.0	84.0	67.0

2	孙悟空	87.5	78.0	77.0
6	孙权	70.0	73.0	78.5
7	宋公明	75.0	65.0	30.0
8	唐三藏	67.0	NULL	80.0

+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

总结:

- 1、order by asc 和 order by 默认是从低到高进行排序的
- 2、order by desc 是从高到低排序的
- 3、对于MySQL的关键字，在用作变量名的时候，一定要加符号：`desc`

注意：查询的顺序问题：

```
mysql> SELECT name, total FROM exam_result
-> ORDER BY chinese + english + math total DESC;
ERROR 1064 (42000): You have an error in your SQL
syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use
near 'total DESC' at line 2
```

错误原因是：他会首先执行：SELECT name, total FROM exam_result；但是此时并不认识 total 是啥？？？

```
mysql> SELECT name, chinese + english + math
total FROM exam_result
-> ORDER BY total DESC;
+-----+-----+
| name    | total |
+-----+-----+
| 猪悟能  | 276.0 |
| 孙悟空  | 242.5 |
| 曹孟德  | 233.0 |
| 孙权    | 221.5 |
| 唐三藏  | 221.0 |
| 刘玄德  | 185.5 |
| 宋公明  | 170.0 |
| 唐三藏  | 146.0 |
| 唐三藏  | NULL  |
```



```
+-----+-----+
9 rows in set (0.00 sec)
```

-- 查询同学各门成绩，依次按 数学降序，英语升序，语文升序的方式显示

```
mysql> SELECT name, math, english, chinese FROM
exam_result
-> ORDER BY math DESC, english, chinese;
```

```
+-----+-----+-----+-----+
| name   | math | english | chinese |
+-----+-----+-----+-----+
| 唐三藏 | 98.0 | 56.0    | 67.0    |
| 猪悟能 | 98.0 | 90.0    | 88.0    |
| 刘玄德 | 85.0 | 45.0    | 55.5    |
| 曹孟德 | 84.0 | 67.0    | 82.0    |
| 孙悟空 | 78.0 | 77.0    | 87.5    |
| 孙权   | 73.0 | 78.5    | 70.0    |
| 宋公明 | 65.0 | 30.0    | 75.0    |
| 唐三藏 | -1.0 | 80.0    | 67.0    |
| 唐三藏 | NULL | 80.0    | 67.0    |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

- where 条件查询

-- 查询数学成绩大于80分的 同学的个人信息及数学成绩

```
mysql> select id,name,math from exam_result where
math > 80;
```

```
+-----+-----+-----+
| id   | name   | math |
+-----+-----+-----+
| 1    | 唐三藏 | 98.0 |
| 3    | 猪悟能 | 98.0 |
| 4    | 曹孟德 | 84.0 |
| 5    | 刘玄德 | 85.0 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

-- 查询数学成绩大于80分的 同学的个人信息及数学成绩,
然后根据数学成绩升序排序

```
mysql> select id,name,math from exam_result where  
math > 80 order by math;  
+-----+-----+-----+  
| id    | name    | math  |  
+-----+-----+-----+  
|      4 | 曹孟德  | 84.0  |  
|      5 | 刘玄德  | 85.0  |  
|      1 | 唐三藏  | 98.0  |  
|      3 | 猪悟能  | 98.0  |  
+-----+-----+-----+  
4 rows in set (0.00 sec)
```

-- 查询数学成绩等于98分的同学及个人信息

```
mysql> select id,name,math from exam_result where  
math = 98;  
+-----+-----+-----+  
| id    | name    | math  |  
+-----+-----+-----+  
|      1 | 唐三藏  | 98.0  |  
|      3 | 猪悟能  | 98.0  |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

注意NULL: = 不安全 对于NULL 查不到

```
mysql> select id,name,math from exam_result where  
math = NULL;  
Empty set (0.00 sec)
```

解决问题: 使用 <=> 这个运算符

```
mysql> select id,name,math from exam_result where  
math <=> NULL;  
+-----+-----+-----+  
| id    | name    | math  |  
+-----+-----+-----+  
|      8 | 唐三藏  | NULL  |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

不等于运算符:

```
mysql> select id,name,math from exam_result where  
math != 98;
```

id	name	math
2	孙悟空	78.0
4	曹孟德	84.0
5	刘玄德	85.0
6	孙权	73.0
7	宋公明	65.0
9	唐三藏	-1.0

6 rows in set (0.00 sec)

```
mysql> select id,name,math from exam_result where  
math <> 98;
```

id	name	math
2	孙悟空	78.0
4	曹孟德	84.0
5	刘玄德	85.0
6	孙权	73.0
7	宋公明	65.0
9	唐三藏	-1.0

6 rows in set (0.00 sec)

遗留问题, 为什么math为NULL的没有显示?? 留作作业!!!!

-- 查询数学成绩70-90之间的同学及成绩
【84-98】

```
mysql> select id,name,math from exam_result where  
math between 84 and 98;
```

id	name	math
----	------	------

1	唐三藏	98.0
3	猪八戒	98.0
4	曹孟德	84.0
5	刘玄德	85.0

```

+-----+-----+-----+
4 rows in set (0.00 sec)

```

```
mysql> select id,name,math from exam_result where
math >= 80 and math<=98;
```

id	name	math
1	唐三藏	98.0
3	猪八戒	98.0
4	曹孟德	84.0
5	刘玄德	85.0

```

+-----+-----+-----+
4 rows in set (0.00 sec)

```

-- 查询数学成绩在98,85,73 中的 所有同学的数学成绩和个人信息

```
mysql> select id,name,math from exam_result where
math in(98,73,85);
```

id	name	math
1	唐三藏	98.0
3	猪八戒	98.0
5	刘玄德	85.0
6	孙权	73.0

```

+-----+-----+-----+
4 rows in set (0.00 sec)

```

in : math in(98,73,85) 代表的是：只要数学成绩属于其中的某一个就进行查询。

```
mysql> select id,name,math from exam_result where
math =98 or math = 73 or math=85;
```

```

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

```

id	name	math
1	唐三藏	98.0
3	猪悟能	98.0
5	刘玄德	85.0
6	孙权	73.0

4 rows in set (0.00 sec)

此时使用的是运算符：or

IS NULL：

IS NOT NULL

```
mysql> select id,name,math from exam_result where
math is NULL;
```

id	name	math
8	唐三藏	NULL

1 row in set (0.00 sec)

```
mysql> select id,name,math from exam_result where
math is NOT NULL;
```

id	name	math
1	唐三藏	98.0
2	孙悟空	78.0
3	猪悟能	98.0
4	曹孟德	84.0
5	刘玄德	85.0
6	孙权	73.0
7	宋公明	65.0
9	唐三藏	-1.0

8 rows in set (0.00 sec)

模糊查询：LIKE like you 立刻有

-- 查找名字当中 包含 "三"的 学生的信息

```
mysql> select * from exam_result where name='三';  
Empty set (0.00 sec)
```

```
mysql> select * from exam_result where name like '%  
三%';
```

```
+-----+-----+-----+-----+-----+  
| id    | name   | chinese | math  | english |  
+-----+-----+-----+-----+-----+  
| 1     | 唐三藏 | 67.0    | 98.0  | 56.0    |  
| 8     | 唐三藏 | 67.0    | NULL  | 80.0    |  
| 9     | 唐三藏 | 67.0    | -1.0  | 80.0    |  
| 10    | 张三丰 | 67.0    | 11.0  | 80.0    |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

```
mysql> select * from exam_result where name like '%  
三'; -- 代表: 一定是以 三这个字 结尾的  
Empty set (0.00 sec)
```

```
mysql> select * from exam_result where name like  
'三%'; -- 代表: 一定是以 三这个字 开始的  
Empty set (0.00 sec)
```

%代表的就是一个通配符。

```
mysql> select * from exam_result where name like  
'唐%';
```

```
+-----+-----+-----+-----+-----+  
| id    | name   | chinese | math  | english |  
+-----+-----+-----+-----+-----+  
| 1     | 唐三藏 | 67.0    | 98.0  | 56.0    |  
| 8     | 唐三藏 | 67.0    | NULL  | 80.0    |  
| 9     | 唐三藏 | 67.0    | -1.0  | 80.0    |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

还有一种模糊匹配:

```
mysql> select * from exam_result where name like '孙_';
```

id	name	chinese	math	english
6	孙权	70.0	73.0	78.5

```
1 row in set (0.00 sec)
```

找到以孙开头的名字，但是记住一个下划线对应一个模糊匹配的字。

```
mysql> select * from exam_result where name like '孙__';
```

id	name	chinese	math	english
2	孙悟空	87.5	78.0	77.0

```
1 row in set (0.00 sec)
```

这个只会找到以孙开头的 且有3个字的 后两个字是模糊匹配的。

NOT 是结合 使用的。

```
mysql> select id,name,math from exam_result where math in(98,73,85);
```

id	name	math
1	唐三藏	98.0
3	猪悟能	98.0
5	刘玄德	85.0
6	孙权	73.0

```
4 rows in set (0.00 sec)
```

```
mysql> select id,name,math from exam_result where math not in(98,73,85);
```

```

+-----+-----+-----+
| id    | name    | math  |
+-----+-----+-----+
| 2     | 孙悟空  | 78.0  |
| 4     | 曹孟德  | 84.0  |
| 7     | 宋公明  | 65.0  |
| 9     | 唐三藏  | -1.0  |
| 10    | 张三丰  | 11.0  |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

练习:

```

DROP TABLE IF EXISTS exam_result2;
CREATE TABLE exam_result2 (
    id INT,
    name VARCHAR(20),
    chinese DECIMAL(3,1),
    math DECIMAL(3,1),
    english DECIMAL(3,1),
    qq_mail varchar(20)
);
INSERT INTO exam_result2 (id,name, chinese, math,
english,qq_mail) VALUES
(1,'唐三藏', 67, 98, 56,'1231@qq.com'),
(2,'孙悟空', 87.5, 78, 77,'1232@qq.com'),
(3,'猪悟能', 88, 98, 90,'1233@qq.com'),
(4,'曹孟德', 82, 84, 67,NULL),
(5,'刘玄德', 55.5, 85, 45,'1234@qq.com'),
(6,'孙权', 70, 73, 78.5,'12345@qq.com'),
(7,'宋公明', 75, 65, 30,'12346@qq.com');

```

- 查询英语不及格的同学及英语成绩 (< 60)
- 查询语文成绩好于英语成绩的同学
- 查询总分在 200 分以下的同学
- 查询语文成绩大于80分, 且英语成绩大于80分的同学
- 查询语文成绩在 [80, 90] 分的同学及语文成绩
- 查询数学成绩是 58 或者 59 或者 98 或者 99 分的同学及数学成绩
- 查询 qq_mail 已知的同学姓名

```
-- 查询英语不及格的同学及英语成绩 ( < 60 )
```



```
select id,name,english from exam_result where
english < 60;

-- 查询语文成绩好于英语成绩的同学
select id,name,english from exam_result where
chinese > english ;

-- 查询总分在 200 分以下的同学  where 条件后面 不可以使用别名
select id,name,chinese+english+math total from
exam_result where chinese+english+math < 200 ;

-- 查询语文成绩大于80分，且英语成绩大于80分的同学
select id,name,english,chinese from exam_result
where chinese > 80 and english > 80 ;

-- 查询语文成绩在 [80, 90] 分的同学及语文成绩
select id,name,chinese from exam_result where
chinese >= 80 and chinese <= 90 ;

select id,name,chinese from exam_result where
chinese between 80 and 90 ;

-- 查询数学成绩是 58 或者 59 或者 98 或者 99 分的同学及
数学成绩
select id,name,chinese from exam_result where math
in(58,59,98,99) ;

-- 查询 qq_mail 已知的同学姓名
select * from exam_result where qq_mail is not NULL;
```

• 分页查询Limit

博哥为何要分页???

答案：主要是有时候，数据量太大了，那么一次性查找数据的时候，系统会执行SQL语句，此时查询是需要时间的。有可能系统就会被卡住！所以，一般优化的方案就是用分页查询。

原理：每次只查询，当前页需要显示的数据即可。如果每页10条数据，那么只查询10条数据。每次点击下一页的

时候，又会请求 查询10条数据。这样 效率就提高了!!!!

语法: select * from exam_result limit s,n; 从偏移位置为s的地方，取出n条数据。

```
mysql> select * from exam_result;
```

	id	name	chinese	math	english
0	1	唐三藏	67.0	98.0	56.0
	2	孙悟空	87.5	78.0	77.0
	3	猪悟能	88.0	98.0	90.0
	4	曹孟德	82.0	84.0	67.0
	5	刘玄德	55.5	85.0	45.0
	6	孙权	70.0	73.0	78.5
	7	宋公明	75.0	65.0	30.0
	8	唐三藏	67.0	NULL	80.0
	9	唐三藏	67.0	-1.0	80.0
	10	张三丰	67.0	11.0	80.0

10 rows in set (0.00 sec)

```
mysql> select * from exam_result limit 0,5;
```

	id	name	chinese	math	english
	1	唐三藏	67.0	98.0	56.0
	2	孙悟空	87.5	78.0	77.0
	3	猪悟能	88.0	98.0	90.0
	4	曹孟德	82.0	84.0	67.0
	5	刘玄德	55.5	85.0	45.0

5 rows in set (0.00 sec)

```
mysql> select * from exam_result limit 5,5;
```

	id	name	chinese	math	english
	6	孙权	70.0	73.0	78.5
	7	宋公明	75.0	65.0	30.0
	8	唐三藏	67.0	NULL	80.0

```
|      9 | 唐三藏 |      67.0 | -1.0 |      80.0 |
|     10 | 张三丰 |      67.0 | 11.0 |      80.0 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

如果n的值过大，那么也没关系，只会查询到，能查询的内容：

```
mysql> select * from exam_result limit 5,7;
+-----+-----+-----+-----+-----+
| id    | name   | chinese | math  | english |
+-----+-----+-----+-----+-----+
|      6 | 孙权   |      70.0 | 73.0 |      78.5 |
|      7 | 宋公明 |      75.0 | 65.0 |      30.0 |
|      8 | 唐三藏 |      67.0 | NULL  |      80.0 |
|      9 | 唐三藏 |      67.0 | -1.0 |      80.0 |
|     10 | 张三丰 |      67.0 | 11.0 |      80.0 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

如果S过大，那么SQL不会报错，但是什么也查不到。

```
mysql> select * from exam_result limit 15,7;
Empty set (0.00 sec)
```

如果没有写S，那么默认是从0偏移开始取n个。

```
mysql> select * from exam_result limit 7;
+-----+-----+-----+-----+-----+
| id    | name   | chinese | math  | english |
+-----+-----+-----+-----+-----+
|      1 | 唐三藏 |      67.0 | 98.0 |      56.0 |
|      2 | 孙悟空 |      87.5 | 78.0 |      77.0 |
|      3 | 猪悟能 |      88.0 | 98.0 |      90.0 |
|      4 | 曹孟德 |      82.0 | 84.0 |      67.0 |
|      5 | 刘玄德 |      55.5 | 85.0 |      45.0 |
|      6 | 孙权   |      70.0 | 73.0 |      78.5 |
|      7 | 宋公明 |      75.0 | 65.0 |      30.0 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

-- 从 s 开始，筛选 n 条结果，比第二种用法更明确，建议使用

SELECT ... FROM table_name [WHERE ...] [ORDER BY ...] **LIMIT**
n OFFSET s;

- 更新update

-- 将孙权的数学成绩，更新为99分。

```
mysql> select * from exam_result;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
2	孙悟空	87.5	78.0	77.0
3	猪悟能	88.0	98.0	90.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
6	孙权	70.0	73.0	78.5
7	宋公明	75.0	65.0	30.0
8	唐三藏	67.0	NULL	80.0
9	唐三藏	67.0	-1.0	80.0
10	张三丰	67.0	11.0	80.0

10 rows in set (0.00 sec)

```
mysql> update exam_result set math=99 where name='孙权';
```

Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> select * from exam_result;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
2	孙悟空	87.5	78.0	77.0
3	猪悟能	88.0	98.0	90.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
6	孙权	70.0	99.0	78.5
7	宋公明	75.0	65.0	30.0
8	唐三藏	67.0	NULL	80.0

9	唐三藏	67.0	-1.0	80.0
10	张三丰	67.0	11.0	80.0

10 rows in set (0.00 sec)

-- 将曹孟德同学的数学成绩变更为 60 分，语文成绩变更为 70 分

UPDATE exam_result SET math = 60, chinese = 70
WHERE name = '曹孟德';

```
mysql> select * from exam_result;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
2	孙悟空	87.5	78.0	77.0
3	猪悟能	88.0	98.0	90.0
4	曹孟德	82.0	84.0	67.0
5	刘玄德	55.5	85.0	45.0
6	孙权	70.0	99.0	78.5
7	宋公明	75.0	65.0	30.0
8	唐三藏	67.0	NULL	80.0
9	唐三藏	67.0	-1.0	80.0
10	张三丰	67.0	11.0	80.0

10 rows in set (0.00 sec)

```
mysql> UPDATE exam_result SET math = 60, chinese = 70 WHERE name = '曹孟德';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from exam_result;
```

id	name	chinese	math	english
1	唐三藏	67.0	98.0	56.0
2	孙悟空	87.5	78.0	77.0
3	猪悟能	88.0	98.0	90.0
4	曹孟德	70.0	60.0	67.0

5	刘玄德	55.5	85.0	45.0
6	孙权	70.0	99.0	78.5
7	宋公明	75.0	65.0	30.0
8	唐三藏	67.0	NULL	80.0
9	唐三藏	67.0	-1.0	80.0
10	张三丰	67.0	11.0	80.0

10 rows in set (0.00 sec)

-- 将总成绩倒数前三的 3 位同学的数学成绩加上 30 分

1、找总成绩倒数前3的同学。

```
mysql> select id ,name,chinese+english+math from
exam_result order by chinese+english+math asc limit
1,3;
```

id	name	chinese+english+math
9	唐三藏	146.0
10	张三丰	158.0
7	宋公明	170.0

3 rows in set (0.00 sec)

2、数学成绩加上 30 分，实际上是更新

update exam_result set math=math+30 order by
chinese+english+math limit 1,3;

```
UPDATE exam_result SET math = math + 30 ORDER BY
chinese + math + english LIMIT 3;
```

