0415_CURD和内置函数

指定列查询 这个是上节课准备好的表

select后面的东西,不仅仅可以跟列名 还可以跟表达式

as可以省略

如果要带上条件呢? 要用where子句

注意,空串和NULL是不同的 另外NULL一般是不参与计算的

运算符	说明
>, >=, <, <=	大于,大于等于,小于,小于等于
=	等于,NULL 不安全,例如 NULL = NULL 的结果是 NULL
<=>	等于, NULL 安全, 例如 NULL <=> NULL 的结果是 TRUE(1)
!=, <>	不等于
BETWEEN a0 AND a1	范围匹配,[a0, a1],如果 a0 <= value <= a1,返回 TRUE(1)
IN (option,)	如果是 option 中的任意一个,返回 TRUE(1)
IS NULL	是 NULL
IS NOT NULL	不是 NULL
LIKE	模糊匹配。%表示任意多个(包括0个)任意字符;_表示任意一个字符

怎么把null筛出来呢

<=>

等于, NULL 安全, 例如 NULL <=> NULL 的结果是 TRUE(1)

IS NULL 是 NULL 用这三种方法是可以的 IS NOT NULL 不是 NULL

这个between语法

是[]区间

这样查也可以

```
MySQL@<lesson6> $ select 1 in (1,2,3,4,5);
+------+
| 1 in (1,2,3,4,5) |
+------+
| 1 |
| 1 |
+-----+
1 row in set (0.00 sec)
```

看这个 1 是不是在后面这个集合里面

查查谁的数学成绩是在这个集合里面的

like 用于模糊匹配

找到姓孙的同学

LIKE

模糊匹配。%表示任意多个(包括0个)任意字符;_表示任意一个字符

```
mysql> select name, id, chinese, english from exam result where chinese > english;
            id | chinese | english
  name
                                       查找语文比英语高的
  唐三藏
                     67
                               56
 孙悟空
             2
                     87
             4
                     82
                               67
 刘玄德
             5
                     55
                               45
                     75
5 rows in set (0.00 sec)
```

现在要找一个总分在200分以下的

mysql> select name,id,chinese+math+english as total from exam_result where tota ERROR 1054 (42S22): Unknown column 'total' in 'where clause' mysql>

为什么这样写不行?因为是先执 行where的 where 不认识total

我们只能这样写

语文成绩大于80并且不姓孙

孙某同学, 否则要求总成绩 > 200 并且 语文成绩 < 数学成绩 并且 英语成绩 > 80

```
mysql> select *,chinese+math+english as total from exam_result where ((english+mat
h+chinese>200) and (chinese<math) and (english>80)) or (name like '孙');
                                  english |
                  chinese
                           math
                                             total
      name
      猪悟能
                                        90
                       88
                              98
                                               276
                                        78
                                               221
2 rows in set (0.00 sec)
mysql>
```

对结果进行排序

6.2.3 结果排序

语法:

```
-- ASC 为升序 (从小到大)
-- DESC 为降序 (从大到小)
-- 默认为 ASC

SELECT ... FROM table_name [WHERE ...]
ORDER BY column [ASC|DESC], [...];
```

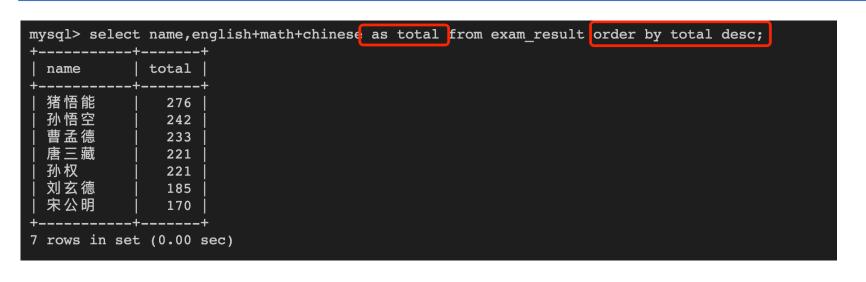
6.2.3.2 同学及 qq 号, 按 qq 号排序显示

-- NULL 视为比任何值都小,升序出现在最上面

用order by 带上 desc和asc

```
mysql> select name, math from exam result order by math asc;
              math
  name
 宋公明
 孙权
                73
                78
                84
                98
7 rows in set (0.00 sec)
mysql> select name, math from exam result order by math desc;
              math
  name
                98
                98
                85
                84
                78
 孙权
                73
 rows in set (0.00 sec)
```

```
mysql> select name, chinese, math, english from exam result order by math desc, english asc, chinese asc;
             chinese
                      math
                             english
  name
                                                    越靠近order就按照谁先排序
  唐三藏
                 67
                        98
                                  56
 猪悟能
                 88
                        98
                                  90
 刘玄德
                 55
                        85
                                  45
 曹孟德
                                  67
                 82
                        84
 孙悟空
                 87
                        78
                                  77
 孙 权
                                  78
                 70
                        73
 宋公明
                 75
                        65
                                  30
7 rows in set (0.00 sec)
```



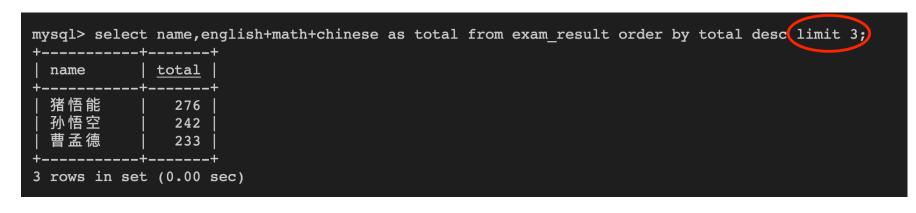
现在问题来了,为什么这里就可以用total呢? 不是说从右到左吗?

为什么where子句不能这里可以呢?

所谓的排序,是不是不需先要有结果?是的排序基本是在select的最后才做的,尽管他的sql语句写在靠右边的位置

6.2.3.5 查询姓孙的同学或者姓曹的同学数学成绩,结果按数学成绩由高到低显示

筛选总分前三的同学



limit一般也是更加靠后去执行 所以先limit出来再去order

所以先limit出来再去order 是不行的

```
-- 起始下标为 0

-- 从 0 开始, 筛选 n 条结果
SELECT ... FROM table_name [WHERE ...] [ORDER BY ...] LIMIT n;

-- 从 s 开始, 筛选 n 条结果
SELECT ... FROM table_name [WHERE ...] [ORDER BY ...] LIMIT s, n;

-- 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;
```

建议:对未知表进行查询时,最好加一条 LIMIT 1,避免因为表中数据过大,查询全表数据导致数据库卡死

现在想要分页显示, 把所有同学按照总分分成3类 优良中

6.3 Update

语法:

```
UPDATE table_name SET column = expr [, column = expr ...]
[WHERE ...] [ORDER BY ...] [LIMIT ...]
```

```
mysql> select name, math from exam result;
 唐三藏
               98
 孙悟空
              80
 猪悟能
              98
              84
 刘玄德
              85
 孙权
              73
 宋公明
7 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 (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from exam result;
                 chinese | math | english
      唐三藏
                                      56
    | 孙悟空
                             80
                                      77
                                      90
                      88
                      70
                                      67
      刘玄德
                                      45
                      55
                            85
                      70
                             73
                                      78
                             65
7 rows in set (0.00 sec)
```

```
mysql> select name, math from exam result;
             math
  name
  唐三藏
               98
  孙悟空
               78
               98
               84
               85
               73
  宋公明
               65
7 rows in set (0.00 sec)
mysgl> update exam result set math=80 where name='孙悟空'
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select name, math from exam result;
             math
  name
  唐三藏
               98
  孙悟空
               80
               98
               84
  刘玄德
               85
               73
  宋公明
               65
7 rows in set (0.00 sec)
mysql>
```

6.3.3 将总成绩倒数前三的 3 位同学的数学成绩加上 30 分

```
mysql> select name,english+math+chinese as total from exam result order by total asc limit 3;
             total
               170
 刘玄德
               185
 曹孟德
               197
3 rows in set (0.00 sec)
mysql> select * from exam result;
                  chinese | math | english
      孙悟空
                             80
                             98
                                       90
                            60
                      70
                                       67
                      55
                             85
                                       45
      刈幺德
                      70
                             73
                                       78
                             65
                                       30
7 rows in set (0.00 sec)
```

```
mysql> update exam result set math=math+30 order by english+math+chinese as asc limit 3;
ERROR 1064 (42000): You have an error in your sol syntax; check the manual that corresponds to your MySQL server
version for the right syntax to use near 'as asc limit 3' at line 1
mysql> update exam result set math=math+30 order by english+math+chinese asc limit 3;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3 Changed: 3 Warnings: 0
mysql> select * from exam result;
                  chinese | math | english
      唐三藏
                                        56
                              98
      孙悟空
                                        77
                              80
      猪悟能
                              98
                                        90
      曹孟德
                             90
                                        67
      刘玄德
                       55
                             115
                                        45
      孙 权
                       70
                             73
                                        78
      宋公明
                             95
                                        30
7 rows in set (0.00 sec)
```

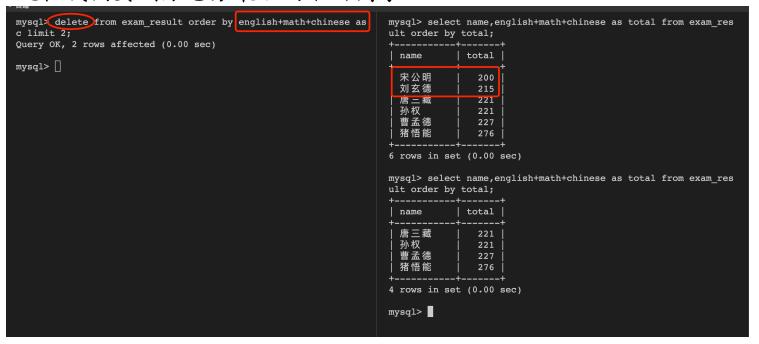
如果我们想给总分+30分呢?

注意: update只能改表内数据! total 不在表里面

删除数据



现在我们要删除总分最后两名的同学



```
mysql> create table if not exists for delete(
   -> id int unsigned primary key auto increment,
   -> name varchar(20) not null
   -> );
Query OK, 0 rows affected (0.02 sec)
mysql> desc for delete
ERROR 1064 (42000): You have an error in your SQL syntax; check the ma
nual that corresponds to your MySQL server version for the right synta
x to use near 'delete' at line 1
mysql> desc for delete;
 Field | Type
                            Null
                                         Default | Extra
                                   Key
         int(10) unsigned
                                                   auto increment
  id
                            NO
                                    PRI |
                                         NULL
         varchar(20)
  name
                            NO
                                         NULL
2 rows in set (0.00 sec)
```

整个表删了 但是 auto_increment没有变

创建个表用来测试删除

```
mysql> select * from for delete;
+----+
  id | name
  2
  3
      C
      d
4 rows in set (0.00 sec)
mysql> delete from for delete;
Query OK, 4 rows affected (0.00 sec)
mysql> select * from for delete;
                               删掉整个表里面的数据
Empty set (0.00 sec)
mysql> show create table for delete\G;
Table: for delete
Create Table: CREATE TABLE `for delete` (
  'id' int(10) unsigned NOT NULL AUTO INCREMENT,
  `name` varchar(20) NOT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO INCREMENT=5 DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
ERROR:
No query specified
mysql>
```

6.4.2 截断表

语法:

```
TRUNCATE [TABLE] table_name
```

```
mysql> insert into for truncate (name) values('A'),('B'),('C');
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> truncate for_truncate;
Query OK, U rows affected (U.U1 sec)
mysql> select * from for_truncate;
Empty set (0.00 sec)
mysql> show create table for truncate;
 Table
                Create Table
 for truncate | CREATE TABLE `for truncate`
  id int (10) ungioned NOT NULL AUTO INCORMENT
                                                       门发现 auto_increment
  `name` varchar(20) NOT NULL,
 PRIMARY KEY ('id')
 ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
                                                               6.4.2 截断表
mysql>
                                                               语法:
```

所以清空表可以用delete/truncate 不过truncate会重置auto_increment truncate只能对整个表操作 TRUNCATE [TABLE] table_name

注意: 这个操作慎用

- 1. 只能对整表操作,不能像 DELETE 一样针对部分数据操作;
- 2. 实际上 MySQL 不对数据操作,所以比 DELETE 更快,但是TRUNCATE在删除数据的时候,并不经过真正的事物,所以无法回滚

. 会重置 AUTO_INCREMENT 项

6.5 插入查询结果

语法:

```
INSERT INTO table_name [(column [, column ...])] SELECT ...
```

案例: 删除表中的的重复复记录, 重复的数据只能有一份

```
mysql> create table duplicate_table_backup like duplicate_table;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from duplicate_table_backup;
Empty set (0.00 sec)

mysql>
```

这个可以把表结构备份但是不能备份里面的数据

意思就是,查出来的不要给我打印出来 查出来之后就向另一个表去插入

相当于就是对表作备份

```
mysql> create table duplicate table(
    -> id int,
   -> name varchar(20)
    -> );
Query OK, 0 rows affected (0.01 sec)
mysql> INSERT INTO duplicate_table VALUES
           (100, 'aaa'),
           (100, 'aaa'),
           (200, 'bbb'),
           (200, 'bbb'),
           (200, 'bbb'),
           (300, 'ccc');
Ouerv OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0
mysql> select * duplicate table;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manu
al that corresponds to your MySQL server version for the right syntax to
use near 'duplicate table' at line 1
mysql> select * from duplicate table;
   100 | aaa
   100 | aaa
   200
        bbb
   200
        bbb
   200
        bbb
        CCC
6 rows in set (0.00 sec)
mysql>
```

```
mysql> insert into duplicate table backup select * from duplicate table;
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0
mysql> select * from duplicate table backup;
+----+
 id
        name
                            把查出来的数据插到另一个表里面
  100
        aaa
  100
        aaa
  200
        bbb
  200
        bbb
  200
        bbb
  300
        CCC
6 rows in set (0.00 sec)
```

我们能对表作备份

我们也能将表去重之后再备份