

4、联合查询

笔记本: Java17&18班上课笔记

创建时间: 2021/3/11 18:21

更新时间: 2021/3/11 21:51

作者: 1262913815@qq.com

联合查询

1、回顾:

NOT NULL - 指示某列不能存储 NULL 值。

UNIQUE - 保证某列的每行必须有唯一的值。

DEFAULT - 规定没有给列赋值时的默认值。

PRIMARY KEY - NOT NULL 和 UNIQUE 的结合。确保某列（或两个列多个列的结合）有唯一标识。
一个表当中 只能有一个主键。

什么主键：通过主键可以唯一的标识这条数据。

一个表当中 包含两个主键，这句话是有瑕疵的。正确的说法是：一个表就是一个主键，但是这个主键，可以使两个字段联合来表示的。联合主键。

识，有助于更容易更快速地找到表中的一个特定的记录。

FOREIGN KEY - 保证一个表中的数据匹配另一个表中的值的参照完整性。

CHECK - 保证列中的值符合指定的条件。对于MySQL数据库，对CHECK子句进行分析，但是忽略CHECK子句

设计表：一对一 一对多 多对多

今日知识：

- GROUP BY子句：分组

准备数据:

```
drop table if exists emp;
create table emp(
    id int primary key auto_increment,
    name varchar(20) not null,
    role varchar(20) not null,
    salary numeric(11,2)
);
insert into emp(name, role, salary) values
('鹏哥','讲师', 1000.20),
('高博','讲师', 2000.99),
('老汤','讲师', 999.11),
('静静','班主任', 333.5),
('莎莎姐','班主任', 700.33),
('隔壁老王','市场', 12000.66);
```

```
mysql> select * from emp;
+----+-----+-----+-----+
| id | name   | role   | salary |
+----+-----+-----+-----+
| 1  | 鹏哥   | 讲师   | 1000.20 |
| 2  | 高博   | 讲师   | 2000.99 |
| 3  | 老汤   | 讲师   | 999.11  |
| 4  | 静静   | 班主任 | 333.50  |
| 5  | 莎莎姐 | 班主任 | 700.33  |
| 6  | 隔壁老王 | 市场   | 12000.66 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)
```

-- 查询每个角色的最高工资、最低工资和平均工资

```
mysql> select max(salary),min(salary),avg(salary)
from emp;
+-----+-----+-----+
| max(salary) | min(salary) | avg(salary) |
+-----+-----+-----+
| 12000.66    | 333.50      | 2839.131667 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

正确的写法:

```
mysql> select
role,max(salary),min(salary),avg(salary) from emp
group by role;
+-----+-----+-----+-----+
| role   | max(salary) | min(salary) | avg(salary) |
+-----+-----+-----+-----+
| 市场   | 12000.66    | 12000.66    | 12000.660000 |
| 班主任 | 700.33      | 333.50      | 516.915000 |
| 讲师   | 2000.99     | 999.11      | 1333.433333 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

执行的时候:

mysql> select role,max(salary),min(salary),avg(salary) from emp group by role;

role	max(salary)	min(salary)	avg(salary)
市场	12000.66	12000.66	12000.660000
班主任	700.33	333.50	516.915000
讲师	2000.99	999.11	1333.433333

- having

过滤条件的。

```
mysql> select
role,max(salary),min(salary),avg(salary)
-> from emp group by role where avg(salary) <
1500;
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 'where avg(salary) < 1500' at line 2
```

正确的写法: 显示平均工资低于1500的角色和它的平均工资

```
mysql> select
role,max(salary),min(salary),avg(salary)
-> from emp group by role having avg(salary) <
1500;
```

role	max(salary)	min(salary)	avg(salary)
班主任	700.33	333.50	516.915000
讲师	2000.99	999.11	1333.433333

2 rows in set (0.00 sec)

• 联合查询

- 1、什么是联合查询：就是两张表或者两张以上的表，进行连接查询。
- 2、为什么要联合查询：就是因为，我们所需要的数据，不仅仅是来自于一张表的，他是来自于多张表的。所以，我们要进行联合查询！！
- 3、前置知识：笛卡尔积！！！ 排列组合而已！



女士表						男士表	
ID	name	笛卡尔积				ID	name
1	小花	1	小花	1	张三	1	张三
2	小燕子	1	小花	2	李四	2	李四
3	大宝	1	小花	3	王麻子	3	王麻子
		2	小燕子	1	张三		
		2	小燕子	2	李四		
		2	小燕子	3	王麻子		
		3	大宝	1	张三		
		3	大宝	2	李四		
		3	大宝	3	王麻子		

所有的联合查询，都是从这个笛卡尔积当中去取出数据！！！当然在取数据的时候，一定是要满足某种规则的。这种规则就是我们接下来学习的各种连表查询的方式！！！！

准备数据：

```
drop database if exists test0311;
create database test0311;
use test0311;
drop table if exists classes;
create table classes(
    id int primary key auto_increment,
    name varchar(50),
    `desc` varchar(50)
);
insert into classes(name, `desc`) values
('计算机系2019级1班', '学习了计算机原理、C和Java语言、数据结构和算法'),
('中文系2019级3班', '学习了中国传统文学'),
('自动化2019级5班', '学习了机械自动化');

drop table if exists student;
create table student(
    id int primary key auto_increment,
    sn int,
```

```

        name varchar(30),
        qq_mail varchar(30),
        classes_id int
    );

insert into student(sn, name, qq_mail, classes_id)
values
('09982','黑旋风李逵','xuanfeng@qq.com',1),
('00835','菩提老祖',null,1),
('00391','白素贞',null,1),
('00031','许仙','xuxian@qq.com',1),
('00054','不想毕业',null,1),
('51234','好好说话','say@qq.com',2),
('83223','tellme',null,2),
('09527','老外学中文','foreigner@qq.com',2);

drop table if exists course;
create table course(
    id int primary key auto_increment,
    name varchar(20)
);

insert into course(name) values
('Java'),('中国传统文化'),('计算机原理'),('语文'),('高
阶数学'),('英文');

drop table if exists score;
create table score(
    id int primary key auto_increment,
    score DECIMAL,
    student_id int,
    course_id int
);

insert into score(score, student_id, course_id)
values
-- 黑旋风李逵
(70.5, 1, 1),(98.5, 1, 3),(33, 1, 5),(98, 1, 6),
-- 菩提老祖
(60, 2, 1),(59.5, 2, 5),
-- 白素贞
(33, 3, 1),(68, 3, 3),(99, 3, 5),
-- 许仙
(67, 4, 1),(23, 4, 3),(56, 4, 5),(72, 4, 6),
-- 不想毕业
(81, 5, 1),(37, 5, 5),

```

```
-- 好好说话
(56, 6, 2),(43, 6, 4),(79, 6, 6),
-- tellme
(80, 7, 2),(92, 7, 6);
```

1、内连接。

语法：

select 字段 from 表1 别名1 [inner] join 表2 别名2 on 连接条件 and 其他条件;

select 字段 from 表1 别名1,表2 别名2 where 连接条件 and 其他条件;

-- 查询“许仙”同学的成绩我们可以发现，要求的信息来自于2张表。 学生表 成绩表

```
select stu.sn,stu.name,sco.score,cou.name as 课程名
from student stu
    inner join score sco on stu.id = sco.student_id
    inner join course cou on sco.course_id = cou.id
and stu.id=4;
```

```
select stu.sn,stu.name,sco.score,cou.name as 课程名
from student stu
    inner join score sco on stu.id = sco.student_id
    inner join course cou on sco.course_id = cou.id
and stu.name='许仙';
```

查询结果显示：

```
mysql> select stu.sn,stu.name,sco.score,cou.name as
课程名
-> from student stu inner join score sco on
stu.id = sco.student_id
-> inner join course cou on sco.course_id =
cou.id
-> and stu.id=4;
```

sn	name	score	课程名
31	许仙	67	Java

sn	name	score	课程名
31	许仙	23	计算机原理
31	许仙	56	高阶数学
31	许仙	72	英文

4 rows in set (0.00 sec)

```
mysql> select stu.sn,stu.name,sco.score,cou.name as
课程名 from student stu
-> inner join score sco on stu.id =
sco.student_id
-> inner join course cou on sco.course_id =
cou.id
-> and stu.name='许仙';
```

sn	name	score	课程名
31	许仙	67	Java
31	许仙	23	计算机原理
31	许仙	56	高阶数学
31	许仙	72	英文

4 rows in set (0.00 sec)

建议：多张表联合查询的时候，去找每个表之间的关系。

1、inner可以省略

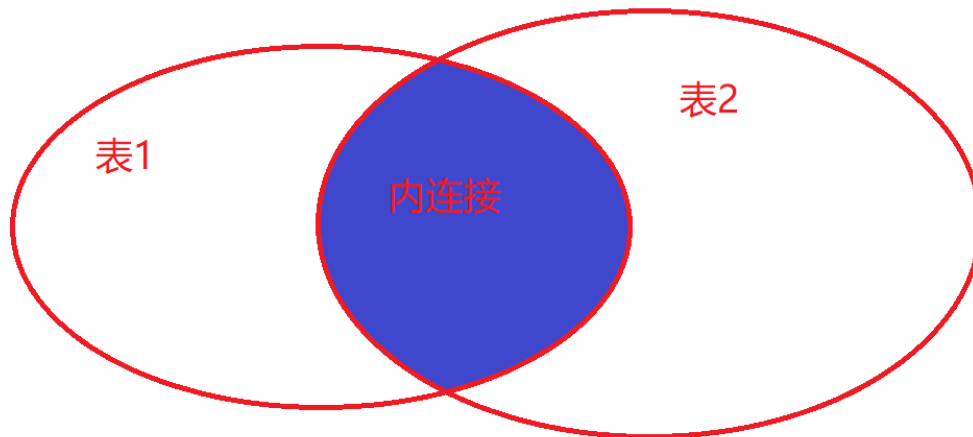
2、或者可以省略join on

```
mysql> select stu.sn,stu.name,sco.score,cou.name as
课程名
-> from student stu ,score sco,course cou
-> where stu.id = sco.student_id
-> and sco.course_id = cou.id
-> and stu.name='许仙';
```

sn	name	score	课程名
31	许仙	67	Java
31	许仙	23	计算机原理
31	许仙	56	高阶数学
31	许仙	72	英文

4 rows in set (0.00 sec)

总结：内连接 就是求几张表的 交集



-- 查询每个同学的总成绩，及同学的个人信息 不用求课程名

```
mysql> select stu.sn,stu.name,sum(score)
-> from student stu,score sco
-> where stu.id = sco.student_id
-> group by stu.id;
```

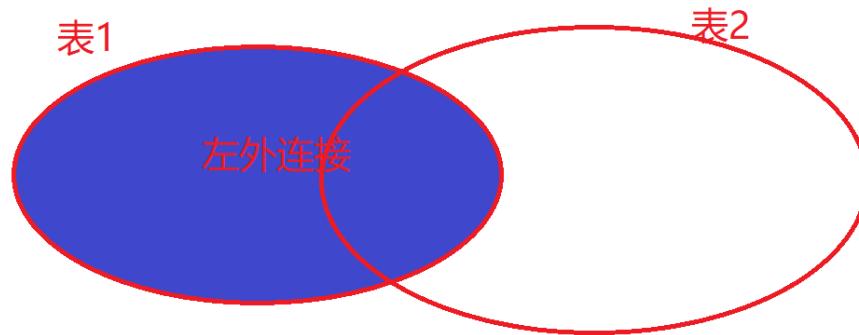
sn	name	sum(score)
9982	黑旋风李逵	301
835	菩提老祖	120
391	白素贞	200
31	许仙	218
54	不想毕业	118
51234	好好说话	178
83223	tellme	172

7 rows in set (0.00 sec)

```
select stu.sn,stu.name,sum(score)
from student stu,score sco
where stu.id = sco.student_id
group by stu.id;
```

2、外连接：左外连接 和 右外连接

左外连接：在进行连接查询的时候，以左边的表为准。



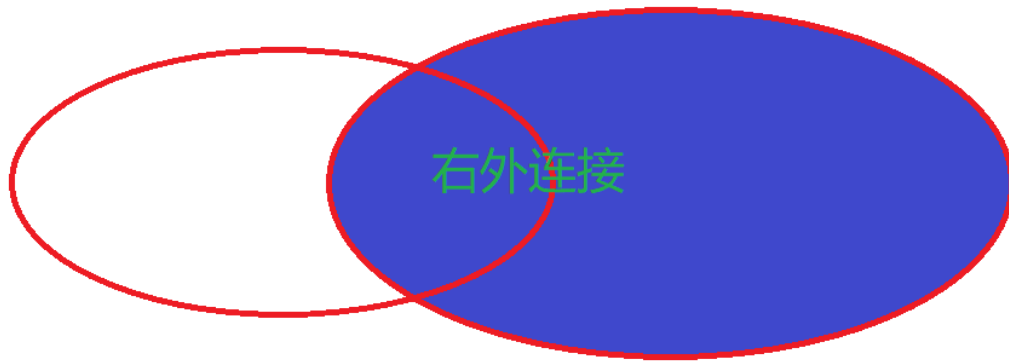
-- 查询每个同学的成绩，及同学的个人信息，如果该同学没有成绩，也需要显示该同学

```
mysql> select stu.id,stu.name,sco.score
-> from student stu left join score sco
-> on stu.id = sco.student_id
-> group by stu.id;
```

id	name	score
1	黑旋风李逵	71
2	菩提老祖	60
3	白素贞	33
4	许仙	67
5	不想毕业	81
6	好好说话	56
7	tellme	80
8	老外学中文	NULL

8 rows in set (0.00 sec)

右外连接：



如果实现上述同样的功能，那么只需要这样修改就好了！！

```
mysql> select stu.id,stu.name,sco.score
-> from score sco right join student stu
-> on stu.id = sco.student_id
-> group by stu.id;
```

id	name	score
1	黑旋风李逵	71
2	菩提老祖	60
3	白素贞	33
4	许仙	67
5	不想毕业	81
6	好好说话	56
7	tellme	80
8	老外学中文	NULL

8 rows in set (0.00 sec)

3、自连接：把一张表 看做两张表来使用。

```
mysql> select * from score;
```

id	score	student_id	course_id
1	71	1	1
2	99	1	3
3	33	1	5
4	98	1	6
5	60	2	1
6	60	2	5
7	33	3	1
8	68	3	3
9	99	3	5
10	67	4	1
11	23	4	3
12	56	4	5
13	72	4	6
14	81	5	1

```
mysql> select * from score;
```

id	score	student_id	course_id
1	71	1	1
2	99	1	3
3	33	1	5
4	98	1	6
5	60	2	1
6	60	2	5
7	33	3	1
8	68	3	3
9	99	3	5
10	67	4	1
11	23	4	3
12	56	4	5
13	72	4	6
14	81	5	1

4、子查询

-- 查询与 “不想毕业” 同学的同班同学

- 单行子查询

1、先查这个人是哪个班的？

```
select classes_id from student stu where  
stu.name='不想毕业';
```

2、查1班有写谁？

```
select * from student where classes_id = 1;
```

3、合并

```
select * from student  
where classes_id = (select classes_id from student  
stu where stu.name='不想毕业');
```

```
mysql> select classes_id from student stu where  
stu.name='不想毕业';
```

```
+-----+
```

```
| classes_id |
```

```
+-----+
```

```
|          1 |
```

```
+-----+
```

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

```
mysql> select * from student where classes_id = 1;
```

```
+-----+-----+-----+-----+-----+
```

```
----+
```

```
| id | sn   | name           | qq_mail           |
```

```
classes_id |
```

```
+-----+-----+-----+-----+-----+
```

```
----+
```

```
| 1 | 9982 | 黑旋风李逵 | xuanfeng@qq.com |
```

```
|          1 |
```

```
| 2 | 835 | 菩提老祖 |
```

```
NULL | | 1 |
```

```
| 3 | 391 | 白素贞 |
```

```
NULL | | 1 |
```

```

| 4 | 31 | 许仙 |
xuxian@qq.com | 1 |
| 5 | 54 | 不想毕业 |
NULL | 1 |
+----+-----+-----+-----+-----+
----+
5 rows in set (0.00 sec)

mysql> select * from student
-> where classes_id = (select classes_id
from student stu where stu.name='不想毕业');
+----+-----+-----+-----+-----+
----+
| id | sn | name | qq_mail |
classes_id |
+----+-----+-----+-----+-----+
----+
| 1 | 9982 | 黑旋风李逵 | xuanfeng@qq.com |
| 1 |
| 2 | 835 | 菩提老祖 |
NULL | 1 |
| 3 | 391 | 白素贞 |
NULL | 1 |
| 4 | 31 | 许仙 |
xuxian@qq.com | 1 |
| 5 | 54 | 不想毕业 |
NULL | 1 |
+----+-----+-----+-----+-----+
----+
5 rows in set (0.01 sec)

```

- 多行子查询：返回多行记录的子查询

【not】in

1、先查询语文 或者英文的ID

```
select id from course where name='语文' or
name='英文';
```

2、去分数表查询这两个ID 对应的成绩

```
select * from score where course_id in (select id
from course where name='语文' or name='英文');
```

```
mysql> select * from score where course_id in
(select id from course where
-> name='语文' or name='英文');
```

id	score	student_id	course_id
4	98	1	6
13	72	4	6
17	43	6	4
18	79	6	6
20	92	7	6

5 rows in set (0.00 sec)

【not】exists: (了解)

你得知道这个关键字执行的原理是什么??

EXISTS(表达式) 只要这个表达式为证 就返回true

```
mysql> select * from student where exists(select id
from student where id=1);
```

id	sn	name	qq_mail	classes_id
1	9982	黑旋风李逵	xuanfeng@qq.com	1
2	835	菩提老祖	NULL	1
3	391	白素贞	NULL	1
4	31	许仙	xuxian@qq.com	1
5	54	不想毕业	NULL	1
6	51234	好好说话	say@qq.com	2
7	83223	tellme	NULL	2
8	9527	老外学中文	foreigner@qq.com	2

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

讲道理：select id from student where id=1 这条SQL语句的查询结果应该是：

```
+-----+
| id |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

那么 你select * from student where exists(select id from student where id=1); 不就和in是一样的吗？打印id为1的学生的所有列的信息吗？

```
+-----+-----+-----+-----+-----+
-----+
| id | sn   | name          | qq_mail          |
classes_id |
+-----+-----+-----+-----+-----+
-----+
| 1 | 9982 | 黑旋风李逵 | xuanfeng@qq.com |
|          1 |
+-----+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)
```

但是 EXISTS(表达式) 只要这个表达式为证 就返回true。

例如这条SQL语句：SELECT * FROM A WHERE EXISTS (SELECT 1 FROM B WHERE B.id = A.id);

- 1、首先执行一次外部查询，并缓存结果集，如 SELECT * FROM A
- 2、遍历外部查询结果集的每一行记录R，代入子查询中作为条件进行查询，如 SELECT 1 FROM B WHERE B.id = A.id

3、如果子查询有返回结果，则EXISTS子句返回TRUE，这一行R可作为外部查询的结果行，否则不能作为结果

实际的例子：

select * from student where exists(select id from student where id=1);

1、select * from student;

```
mysql> select * from student;
+-----+-----+-----+-----+
| id | sn   | name       | qq_mail           |
|-----|-----|-----|-----|
| 1 | 9982 | 黑旋风李逵 | xuanfeng@qq.com   |
| 2 | 835  | 菩提老祖   | NULL              |
| 3 | 391  | 白素贞     | NULL              |
| 4 | 31   | 许仙       | xuxian@qq.com     |
| 5 | 54   | 不想毕业   | NULL              |
| 6 | 51234 | 好好说话   | say@qq.com        |
| 7 | 83223 | tellme     | NULL              |
| 8 | 9527 | 老外学中文 | foreigner@qq.com  |
| 2 |      |            |                   |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

2、
遗留问题

合并查询：

```
mysql> select * from student where id<=3
-> union
-> select * from student where name='白素贞';
```



```

+----+-----+-----+-----+-----+
----+
| id | sn   | name       | qq_mail       |
classes_id |
+----+-----+-----+-----+
----+
|  1 | 9982 | 黑旋风李逵 | xuanfeng@qq.com
|          1 |
|  2 |  835 | 菩提老祖   |
NULL      | 1 |
|  3 |  391 | 白素贞     |
NULL      | 1 |
+----+-----+-----+-----+
----+
3 rows in set (0.01 sec)

```

```

mysql> select * from student where id<=3
-> union
-> select * from student where name='许仙';

```

```

+----+-----+-----+-----+-----+
----+
| id | sn   | name       | qq_mail       |
classes_id |
+----+-----+-----+-----+
----+
|  1 | 9982 | 黑旋风李逵 | xuanfeng@qq.com
|          1 |
|  2 |  835 | 菩提老祖   |
NULL      | 1 |
|  3 |  391 | 白素贞     |
NULL      | 1 |
|  4 |   31 | 许仙       |
xuxian@qq.com | 1 |
+----+-----+-----+-----+
----+
4 rows in set (0.00 sec)

```

```

mysql>
mysql> select * from student where id<=3
-> union all
-> select * from student where name='白素贞';

```

```

+----+-----+-----+-----+-----+
----+
| id | sn   | name          | qq_mail          |
classes_id |
+----+-----+-----+-----+-----+
----+
|  1 | 9982 | 黑旋风李逵   | xuanfeng@qq.com  |
|      1 |
|  2 | 835  | 菩提老祖     |                   |
NULL      |           1 |
|  3 | 391  | 白素贞       |                   |
NULL      |           1 |
|  3 | 391  | 白素贞       |                   |
NULL      |           1 |
+----+-----+-----+-----+-----+
----+
4 rows in set (0.00 sec)

mysql>

```

联系MySQL网站: <https://www.nowcoder.com/ta/sql>