



重构IT职业教育新生态

MySQL

- 查询语句

目录

C O N T E N T S

1

简单查询

2

分组查询和子查询

3

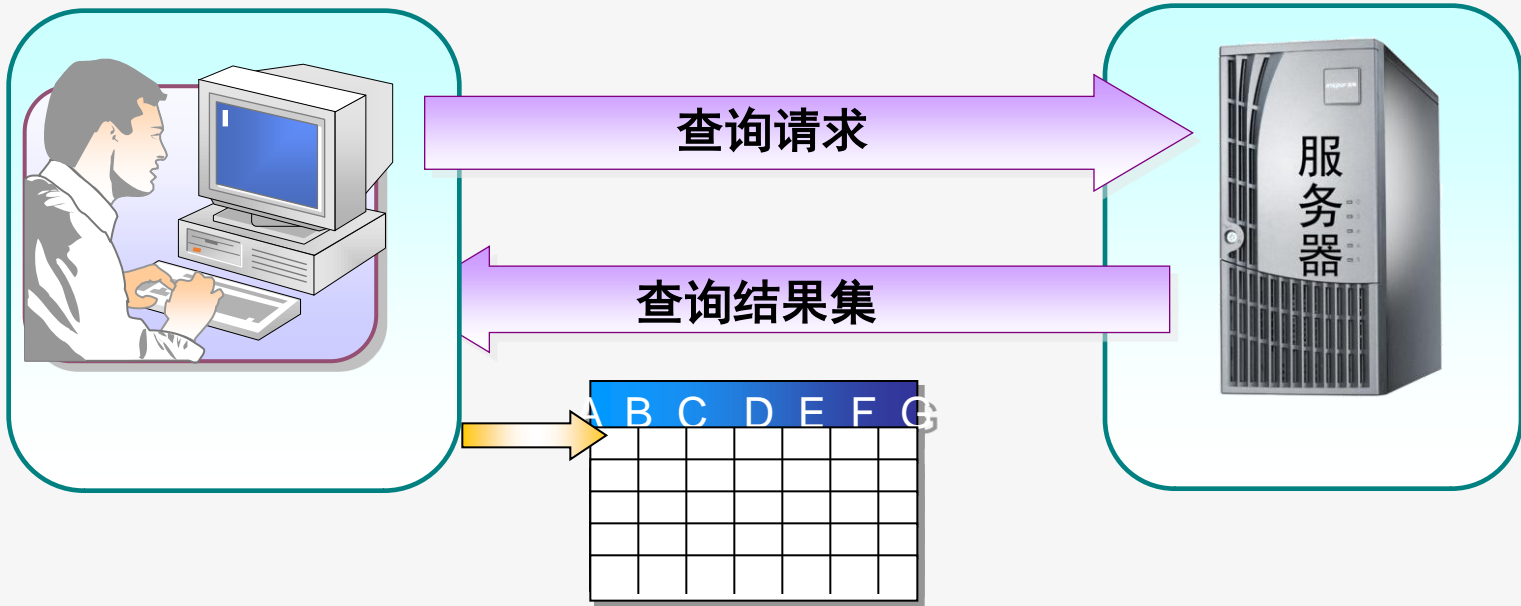
关联查询

本章目标

- 一．熟练掌握基本查询语句
- 二．熟练掌握排序查询
- 三．熟练掌握分组查询与聚合函数
- 四．掌握关联查询
- 五．应用SQL进行综合查询

什么是查询？

```
SELECT * FROM SALES
```



查询产生一个虚拟表

看到的是表形式显示的结果，但结果并不真正存储

每次执行查询只是现从数据表中提取数据，并按照表的形式显示出来

基础查询语法

```
SELECT <列名>
FROM <表名>
[WHERE <查询条件表达式>]
[ORDER BY <排序的列名>[ASC或DESC]]
```

列名称	表名	过滤条件	排序条件
SELECT SCode, SName, SAddress	FROM Students	WHERE SSEX = 0	ORDER BY SCode

```
SELECT  SCode,SName,SAddress
FROM      Students
WHERE  SSEX = 0
ORDER BY  SCode
```

查询全部的行和列

```
SELECT * FROM Students
```

```
SELECT SCode,SName,SAddress FROM Students
```

查询部分列

```
SELECT SName,SAddress FROM Students
```

```
SELECT SCode FROM Students
```

使用AS来命名列

```
SELECT SCode AS 学员编号,SName AS学员姓名,SAddress AS 学员地址  
FROM Students
```

```
SELECT 工资 + 奖金 AS '收入'  
FROM Employees
```

注意：

1. + 连接的数据类型必须兼容
2. + 连接必须为INT型

限制固定行数

```
MYSQL:SELECT SName, SAddress  
FROM Students WHERE SSex = 0 LIMIT 5, 10
```


数据查询-排除重复

使用DISTINCT排除重复

```
SELECT DISTINCT 列A FROM 表
```

使用DISTINCT多列排除重复

注意：
这并不意味着任一列没有重复
只有在组合时没有重复

```
SELECT DISTINCT 列A , 列B FROM 表
```

在公司的员工表里，每个员工都有自己的职位，但职位有相同的，想查看目前一共有多少个职位就需要用到排除重复

单条件选择

```
SELECT 列A...FROM 表 WHERE 条件 = 值
```

```
SELECT name FROM student WHERE id = 1
```

多条件选择

```
SELECT 列A...FROM 表 WHERE 条件 = 值 AND/OR 条件 = 值
```

```
SELECT name FROM student WHERE id = 1 AND pass = '123'
```

SQL中的运算符

运算符	含义
=	等于
>	大于
<	小于
>=	大于或等于
<=	小于或等于
!= (<>)	不等于

逻辑表达式	说明
AND (&&)	逻辑与 A AND B
OR ()	逻辑或 A OR B
NOT (!)	逻辑非 NOT A

案例

- NOT (付款方式 = '信用卡') OR (信用卡 <> '阳光卡')
- 这个表达式的含义是什么？

数据查询-范围操作



范围选择

```
SELECT name FROM student WHERE age >= 18 AND age<= 28
```

BETWEEN

SELECT 列 FROM 表 WHERE 条件 BETWEEN 下限 AND 上限 (上下限的位置不能颠倒)

```
SELECT name FROM student WHERE age BETWEEN 18 AND 28
```

NOT BETWEEN

SELECT 列 FROM 表 WHERE 条件 NOT BETWEEN 下限 AND 上限

```
SELECT name FROM student WHERE age NOT BETWEEN 18 AND 28
```

数据查询-集合操作

匹配多个值

```
SELECT 列A...FROM 表 WHERE 条件1 = 值1 OR 条件1 = 值2 OR 条件1 = 值3
```

IN

```
SELECT 列A...FROM 表 WHERE 条件 IN (值1,值2,值3)
```

```
SELECT name FROM student WHERE id IN (1,2,3)
```

NOT IN

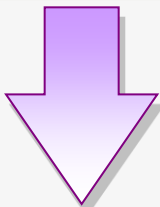
```
SELECT 列A...FROM 表 WHERE 条件 NOT IN (值1,值2,值3)
```

```
SELECT name FROM student WHERE id NOT IN (1,2,3)
```

模糊查询—BETWEEN

- 把某一字段中内容在特定范围内的记录查询出来

```
SELECT StudentID, Score FROM SCore WHERE Score  
BETWEEN 60 AND 80
```



StudentID	Score
7	77
8	61
11	76
17	64
...	...

LIKE

SELECT 列A...FROM 表 WHERE 条件LIKE 通配符

通配符	解释	示例
_	有且仅有一个字符	A LIKE 'C_'
%	任意长度的字符串，0或N个	B LIKE 'CO_%'

数据查询-空行及常量列

查询空行

```
SELECT SName FROM Students WHERE SEmail IS NULL
```

使用常量列

```
SELECT Sname as 姓名, Saddress as 地址, '河北新龙' AS 学校名称  
FROM Students
```


升序排列

```
SELECT StudentID AS 学员编号,(Score*0.9+5) AS 综合成绩  
FROM Score  
WHERE (Score*0.9+5)>60  
ORDER BY Score ASC
```

降序排列

```
SELECT StudentID AS 学员编号,(Score*0.9+5) AS 综合成绩  
FROM Score  
ORDER BY 综合成绩 DESC
```

按多列排序

```
SELECT StudentID AS 学员编号, Score AS 成绩  
FROM Score  
WHERE Score>60  
ORDER BY Score > 60, CourseID
```



- 排序中的字段，可以使用表达式吗？
 - ◆ 如果不可以，请说明原因
 - ◆ 如果可以，请举例说明

- 为从数据库中查询数据，必须使用SQL的SELECT语句。
- 所有的SELECT语句均使用SELECT关键字开头。
- SELECT * FROM <表名>可以返回表中所有列。
- 若表名或列名是一个SQL关键字或是数据库的保留字，则需将表名或列名用中括号或双引号括起来。
- 通过SELECT语句，既可以返回单列数据，也可以返回多列数据。
- 使用别名，可以使SQL语句和返回的数据更易读。
- 在SQL语句中，可以对列执行一般的加减乘除数学计算。
- 在SQL语句中使用DISTINCT关键字可以排除重复数据。
- 在SQL语句中使用WHERE子句来过滤查询，过滤条件可以是多种形式。
- 为对数据排序，可以在SQL语句中使用ORDER BY子句。当ORDER BY子句后只指定一个列名时，即按照单列排序。如果指定多个列名时，即按照多列排序。
- 在ORDER BY子句中的列名上指定ASC或DESC关键字，使数据按照升序或降序排列。如果不指定该关键字，则默认以升序的方式排列。

- 常用聚合函数
 - COUNT：返回结果集中行的数目
 - SUM：返回结果集中所有值的总和
 - AVG：返回结果集中所有值的平均值
 - MAX：返回结果集中所有值的最大值
 - MIN：返回结果集中所有值的最小值

```
SELECT COUNT(<计数规范>) FROM <表名>;
```

- 执行行和列计数
 - 因为这是一个SELECT语句，我们可以在其中包含一个WHERE子句及其它SELECT语句可以使用的选项。如果没有过滤条件，那么所有的行都被COUNT函数处理。



成绩表中存储了所有学员的成绩，想知道：
学员的总成绩、平均成绩、有成绩的学员总共有多少名？
怎么办？

SUM

```
SELECT SUM(ytd_sales) FROM titles WHERE type = 'business'
```

```
SELECT SUM(ytd_sales) ,Price  
FROM titles WHERE type = 'business'
```

AVG

```
SELECT AVG(SCore) AS 平均成绩  
FROM Score WHERE Score >=60
```

注意：MySQL中，SELECT后出现了聚合函数，则该位置只能出现其他聚合函数和分组依据列，普通列只显示该列第一行值。而MSSQL直接报错，不支持该行为。

MAX、MIN

```
SELECT AVG(SCore) AS 平均成绩, MAX (Score) AS 最高分,  
MIN (Score) AS 最低分 FROM Score WHERE Score >=60
```

COUNT

```
SELECT COUNT (*) AS 及格人数 FROM Score  
WHERE Score>=60
```

count(*) 和 count(列)的区别在于
是否忽略null



如果不是统计所有人所有课程的总成绩
而是想求每一门课的平均绩或者某个人的所有课的
总成绩怎么办？

分组汇总

	StudentID	CourseID	Score	ScoreID
1	31	1	95	2
2	29	1	67	5
3	13	1	56	6
4	15	1	81	7
5	17	1	82	8
6	19	1	78	9
7	13	3	81	10
8	15	3	92	12
9	17	3	81	13
10	19	3	66	14
11	29	3	36	15
12	31	3	73	16
13	31	9	57	17
14	29	9	76	18
15	13	9	78	19
16	15	9	89	20
17	19	9	93	21

课程1的6个成绩取平均值

课程3的6个成绩取平均值

课程9的5个成绩取平均值

分组查询—GROUP BY

```
SELECT CourseID, AVG(Score) AS 课程平均成绩  
FROM Score  
GROUP BY CourseID
```



	CourseID	课程平均成绩
1	1	76
2	3	71
3	9	78

问题

- 思考：执行以下的SQL：

```
SELECT StudentID, CourseID, AVG(Score) AS 课程平均成绩  
FROM Score  
GROUP BY CourseID
```

—— 结果如何？

分组查询—多列分组

```
SELECT StudentID AS 学员编号, CourseID AS 内部测试, AVG(Score) AS 内部测试平均成绩  
FROM Score  
GROUP BY StudentID, CourseID
```



	学员编号	内部测试	内部测试平均成绩
1	13	1	68
2	15	1	81
3	17	1	82
4	19	1	78
5	29	1	67
6	31	1	95
7	13	3	81
8	15	3	92
9	17	3	81
10	19	3	66
11	29	3	62
12	31	3	73
13	13	9	78
14	15	9	89
15	19	9	93
16	29	9	76
17	31	9	63



在以上统计测试成绩的基础上，
如果只想看补考的学员的成绩，怎么办

分组查询—再看看

	StudentID	CourseID	Score	ScoreID
1	31	1	95	2
2	29	1	67	5
3	13	1	56	6
4	15	1	89	7
5	17	1	82	8
6	19	1	78	9
7	13	3	81	10
8	15	3	92	12
9	17	3	81	13

SELECT StudentID AS 学员编号, CourseID AS 内部测试, AVG(Score) AS 内部测试平均成绩
FROM Score
GROUP BY StudentID, CourseID

增加条件：要求该学员的CourseID在分组内出现过一次以上.....

12	31	3	73	16
13	31	9	57	17
14	29	9	76	18
15	13	9	78	19
16	15	9	89	20
17	19	9	93	21
18	13	1	80	22
19	29	3	88	23
20	31	9	69	24

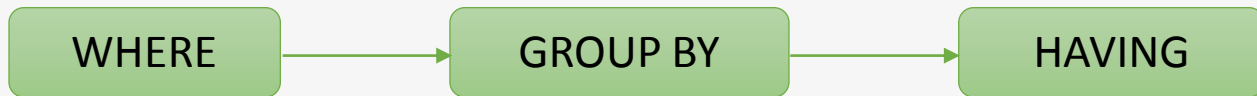
分组查询—HAVING

```
SELECT StudentID AS 学员编号, CourseID AS 内部测试,  
AVG(Score) AS 内部测试平均成绩  
FROM Score  
GROUP BY StudentID, CourseID  
HAVING COUNT(Score) > 1
```



	学员编号	内部测试	内部测试平均成绩
1	13	1	68
2	29	3	62
3	31	9	63

- WHERE子句：
 - 从数据源中去掉不符合其搜索条件的数据
- GROUP BY子句：
 - 搜集数据行到各个组中，统计函数为各个组计算统计值，每个组输出一行结果
- HAVING子句：
 - 从分组计算结果中进行过滤，去掉不符合其组搜索条件的各组数据行





分析以下SQL的含义

```
SELECT  部门编号, COUNT(*)  
FROM    员工信息表  
WHERE   工资 >= 2000  
GROUP BY  部门编号  
HAVING  COUNT(*) > 1
```

- 以下的查询语句：

- SELECT * FROM A

- 将得到以下的查询结果：

- A1 B1 C1
 - 1001 P1 70
 - 1002 P2 60
 - 1001 P1 100
 - 1002 P1 80
 - 1002 P2 90

- 那么，以下的查询语句将输出哪些信息？

```
SELECT A1,B1,MAX(C1) FROM A
```

- 关系型数据库管理系统在执行一条SQL时，按照如下顺序执行各子句。
 - 首先执行FROM子句,从表中查找数据；
 - 如果有WHERE子句，则根据其中的过滤条件，从表中去掉不满足过滤条件的行。
 - 根据GROUP BY子句中指定的分组列，对中间表中的数据进行分组。
 - 为每个组计算SELECT子句聚合函数的值，并为每组生成查询结果中的一行。
 - 如果有HAVING子句，则根据HAVING子句的过滤条件，分组计算聚合计算的结果再次过滤。
 - 如果有ORDER BY子句中，则根据ORDER BY子句中的列，对结果集进行排序。

Select AVG(列名) from tablename

where **

Group by <列名>

Having

ORDER BY <列名> ASC/DESC

limit

什么是子查询？

子查询是指嵌入到另一条SQL 语句中的查询语句。

数据库引擎将子查询做为虚表执行查询操作。子查询可做为连接语句中的一个表，结果可做为选择语句中的一个值

怎么查考试成绩为70分的学生姓名？

```
SELECT 学员姓名 FROM Students WHERE 学员编号= '004'
```

学员编号	学员姓名
001	张明全
002	李菲
003	于寄谦
004	刘国正
005	王
006	李
007	王
008	王
009	王

刘国正

```
SELECT 学员编号 FROM Students WHERE 学员成绩= 70
```

学员编号	学员成绩
001	50
002	60
003	55
004	70
005	7
006	8
007	2
008	1

004

因此，我们是把第二张表的结果做为了第一张表的条件

查询某同学的班级名

```
SELECT name FROM class WHERE id = ?
```

```
SELECT cid FROM students WHERE name = 'xxx'
```

子查询部分

```
SELECT name FROM class WHERE id = (SELECT cid FROM students WHERE name = 'xxxx')
```

```
SELECT <列名>  
FROM <表名>  
[WHERE <SELECT <列名> FROM <表名>]
```

列名称	表名	过滤条件	子查询
SELECT SName,	FROM Students	WHERE Id	SELECT SCode

```
SELECT    SName,SAddress  
FROM      Students  
WHERE     id =  
子查询   SELECT sid FROM mark WHERE mark = 70
```

•子查询在WHERE语句中的一般用法:

- SELECT ... FROM 表 WHERE 过滤字段 (比较运算符) (子查询)
- 外面的查询称为父查询, 括号中嵌入的查询称为子查询
- UPDATE、INSERT、DELETE一起使用, 语法类似于SELECT语句
- 将子查询和比较运算符联合使用, 必须保证子查询返回的值不能多于一个

查询全部女生的名字

```
SELECT name FROM Students WHERE id = (SELECT sid FROM studentinfo WHERE sex = 0)
```

这样查询能不能得到我们想要的结果

查询所有女生的ID

```
SELECT sid FROM studentinfo WHERE sex = 0
```

注意：
这里将返回多条纪录

查询所有女生的name

```
SELECT name FROM students WHERE id IN (SELECT sid FROM studentsinfo WHERE sex = 0)
```

注意：
使用**IN**关键字来匹配子查询中返回的多条纪录

	stuName	stuNo	stuSex	stuAge	stuSeat	stuAddress
1	张秋丽	s25301	男	18	1	北京海淀
2	李文才	s25302	男	31	3	地址不详
3	李斯文	s25303	女	22	2	河南洛阳
4	欧阳俊雄	s25304	男	28	4	新疆
5	梅超风	s25318	女	23	5	地址不详

	ExamNo	stuNo	writtenExam	LabExam
1	s271811	s25303	90	58
2	s271813	s25302	60	90
3	s271816	s25301	87	82

学员信息表和成绩表

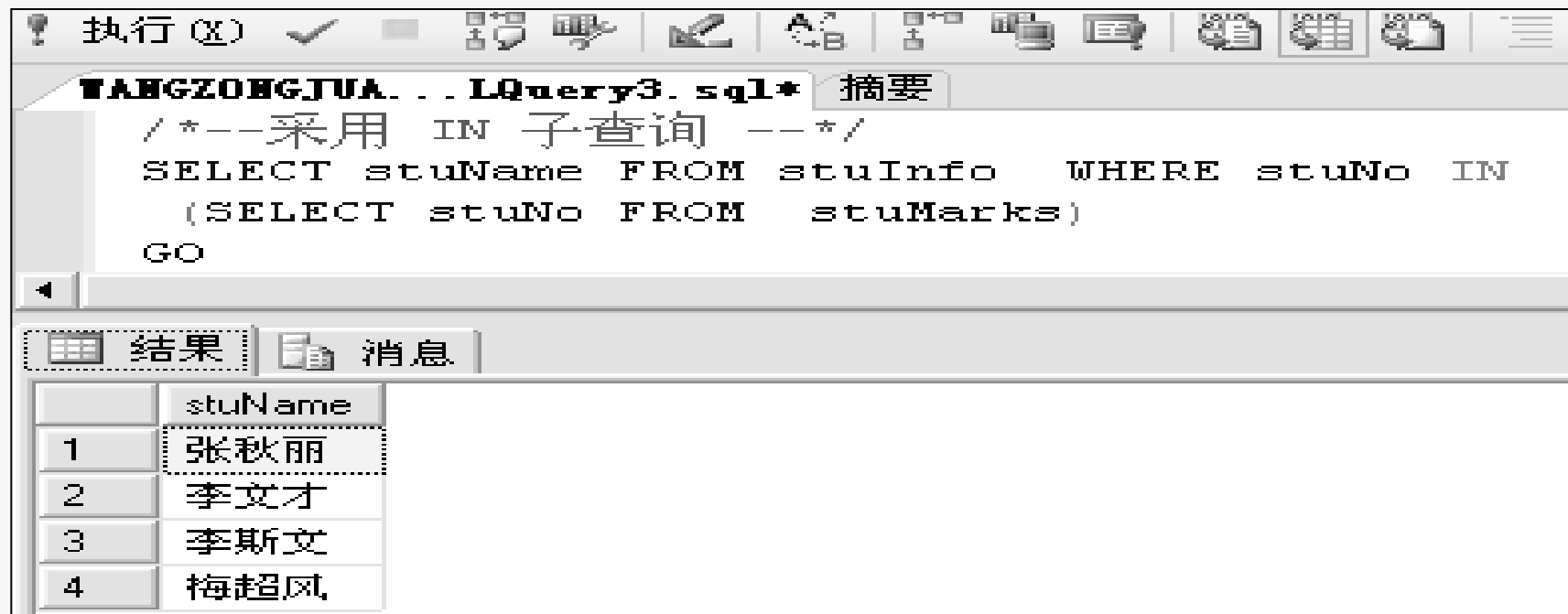
问题

- 查询参加考试的学员名单

分析

- 判断一个学员是否参加考试其实很简单，只需要查看该学员对应的学号是否在考试成绩表stuMarks中出现即可

- 参考语句



The image shows a screenshot of a SQL query editor and its results. The editor window has a title bar with a file icon and the text "TANGZONGJUA...LQuery3.sql". Below the title bar is a toolbar with various icons. The main text area contains the following SQL code:

```
/*--采用 IN 子查询 --*/  
SELECT stuName FROM stuInfo WHERE stuNo IN  
  (SELECT stuNo FROM stuMarks)  
GO
```

Below the editor is a results window with two tabs: "结果" (Results) and "消息" (Messages). The "结果" tab is active, showing a table with the following data:

	stuName
1	张秋丽
2	李文才
3	李斯文
4	梅超风

演示：使用IN 子查询

提问

- 查询未参加考试的学员名单

分析

- 加上否定的NOT 即可

- EXISTS子查询的语法：

```
SELECT 列名 FROM 表名 WHERE EXISTS (子查询)  
语句
```

- 如果子查询的结果非空，即记录条数1条以上，则EXISTS (子查询) 将返回真 (true)，否则返回假(false)
- EXISTS也可以作为WHERE 语句的子查询，但一般都能用IN子查询替换

EXISTS子查询

学员编号	学员姓名
001	张明全
002	李菲
003	于寄谦
004	刘国正
005	周接轮
006	巩小妹
007	巩大妹
008	张明敏
009	矛十八
010	罗林光
011	司马坡



学员编号	学员成绩
001	50
002	60
003	55
004	70
005	77
006	78
007	82
008	65
009	54
010	76
011	72

```
SELECT name FROM students AS s WHERE EXISTS (SELECT * FROM 成绩 AS e WHERE e.ID=s.ID)
```



Students表id分别为001.002.003....011

可以看做

```
SELECT * FROM 成绩 AS e WHERE e.ID = 001
```

```
SELECT * FROM 成绩 AS e WHERE e.ID = 002
```

.....

```
SELECT * FROM 成绩 AS e WHERE e.ID = 011
```

有数据返回true

```
SELECT name FROM students AS s 执行
```

查询所有范姓同学的班级

```
SELECT name FROM class WHERE id IN (SELECT cid FROM students WHERE name LIKE '范%')
```

查询所有范姓同学的班级

```
SELECT name FROM class AS c WHERE EXISTS (SELECT * FROM students AS s WHERE name LIKE '范%' AND s.cid = c.id)
```

查询

```
SELECT id FROM 成绩 WHERE 分数 > (SELECT 分数 FROM 统计)
```

ANY查询

```
SELECT id FROM 成绩 WHERE 分数 > ANY (SELECT 分数 FROM 统计)
```

学员编号	学员成绩
002	60
003	55
004	70
005	77

ANY子句和ALL子句

- 1.ANY子句表示父查询只需要任意匹配一条子查询结果就可执行
- 2.ANY子句中 = ANY(子查询) 与IN同意
- 3.ANY子句中 > ANY代表大于最小值
- 4.ANY子句中 < ANY代表小于最大值

- 1.ALL子句中 > ALL代表大于最大值
- 2.ALL子句中 < ALL代表小于最小值

```
SELECT id FROM 成绩 WHERE 分数 > ALL (SELECT 分数 FROM 统计)
```

学员编号	学员成绩
005	77

学员编号	学员名字
001	范娜
002	张三星
003	李四海
004	刘德华
005	周润发

用户编号	学员名字
001	范娜
002	张学友
003	喜羊羊
004	灰太狼
005	阿童木

UNION子句

学员编号	学员名字
001	范娜
002	张三星
003	李四海
004	刘德华
005	周润发

用户编号	学员名字
001	范娜
002	张学友
003	喜羊羊
004	灰太狼
005	阿童木

编号	学员名字
001	范娜
002	张三星
003	李四海
004	刘德华
005	周润发
006	张学友
007	喜羊羊
008	灰太狼
009	阿童木

1. 集1与集2的交集
2. 去掉集1集2之间重复的部分
3. 重复内容只保留一个
4. 两个集会合并为一个新集返回

```
SELECT name FROM students UNION  
SELECT name FROM 统计
```

UNION子句

学员编号	学员名字
001	范娜
002	张三星
003	李四海
004	刘德华
005	周润发

用户编号	学员名字
001	范娜
002	张学友
003	喜羊羊
004	灰太狼
005	阿童木

1. 集1与集2的并集
2. 不去重复的部分
3. 将集一和集二直接相加
4. 两个集会合并为一个新集返回

编号	学员名字
001	范娜
002	张三星
003	李四海
004	刘德华
005	周润发
006	张学友
007	喜羊羊
008	灰太狼
009	范娜
010	阿童木

```
SELECT name FROM students UNION ALL  
SELECT name FROM 统计
```

- 嵌套在其它查询中的查询称为子查询。
- 子查询又称内部查询，包含子查询的查询语句成为外部查询。
- 在查询中，我们可以使用表的别名和列的别名。当我们使用多个表时，使用表别名的优点是显然的。
- 在有多个子查询的查询，每个组成子查询的查询仅执行一次，每个查询将传递一个值或者一系列值到一个外部查询。
- EXIST运算符允许查找满足特定条件的行。EXISTS后面总是跟一个子查询，只要子查询返回了行，EXISTS的值就为真。
- ALL运算符与子查询和比较运算符（例如=、>或>=）一起使用。如果子查询返回的所有值都满足比较运算，那么比较表达式就为真。如果不是所有值都满足比较运算或子查询没有给外部语句返回行，则返回假。
- ANY与ALL的工作方式不同的是：在子查询返回的值中，只要有一行满足比较运算，那么比较表达式就为真。如果所有值都不满足，则返回假。
- SQL允许使用UNION ALL运算符返回每个数据集的所有成员，而不考虑重复。

为什么要表连接

学员编号	学员姓名	CLASSID
001	张明全	1
002	李菲	1
003	于寄谦	1
004	刘国正	1
005	周接轮	1
006	巩小妹	2
007	巩大妹	2
008	张明敏	3
009	矛十八	3
010	罗林光	3
011	司马坡	3

班级编号	班级名
1	T01
2	T02
3	T03

学员姓名	班级名
张明全	T01
李菲	T01
于寄谦	T01
.....

多表联接查询—分类

- 内联接(INNER JOIN)
- 外联接
 - 左外联接 (LEFT JOIN)
 - 右外联接 (RIGHT JOIN)

多表联接查询—内联接

猜一猜：这样写，返回的查询结果是什么？

```
SELECT  S.SName,C.CourseID,C.Score
FROM      Students AS S
INNER JOIN      Score AS C
ON         C.StudentID = S.SCode
```

Students

SName	SCode
梅超风	1
陈玄风	2
陆乘风	3
曲灵风	4

Score

StudentsID	CourseID	Score
1	001	97
2	001	89
2	002	67
3	002	76
3	003	81

查询结果

SName	CourseID	Score
梅超风	001	97
陈玄风	001	89
陈玄风	002	67
陆乘风	002	76
陆乘风	003	81

多表联接查询—内联接



```
SELECT Students.SName, Score.CourseID, Score.Score  
FROM    Students,Score  
WHERE Students.SCode = Score.StudentID
```

注意: 在这个连接语句中, 我们使用逗号来代替了JOIN

多表联接查询—三表联接

SELECT

S.SName AS 姓名, CS.CourseName AS 课程, C.Score AS 成绩

FROM Students AS S

INNER JOIN Score AS C ON (S.SCode = C.StudentID)

INNER JOIN Course AS CS ON (CS.CourseID = C.CourseID)

SELECT

S.SName AS 姓名, CS.CourseName AS 课程, C.Score AS 成绩

FROM Students AS S , Score AS C , Course AS CS

WHERE (S.SCode = C.StudentID) AND (CS.CourseID = C.CourseID)

多表联接查询

```
SELECT  S.SName,C.CourseID,C.Score
FROM      Students AS S
LEFT JOIN  Score AS C
ON        C.StudentID = S.SCode
```

```
SELECT Titles.Title_id, Titles.Title, Publishers.Pub_name
FROM titles
RIGHT JOIN Publishers
ON Titles.Pub_id = Publishers.Pub_id
```

注意: 所谓左右表只是针对JOIN而言, JOIN之前的为左表, JOIN之后的为右表

- left join(左联接) 返回包括左表中的所有记录和右表中联结字段相等的记录
- right join(右联接) 返回包括右表中的所有记录和左表中联结字段相等的记录
- inner join(等值连接) 只返回两个表中联结字段相等的行

- 当一个子查询是相关子查询的时候，会严重影响性能。SQL给我们提供了一种更简单的语法来同时查询多个表，即使用联接。
- 所有的相等联接都在WHERE子句使用相等运算符（=），我们也可以在联接中使用其它比较运算符，区分内联接与外联接及其实践中的应用。
- 有两种形式的外联接方式，左外联接，右外联接。所有的行都包括了左联接表，右联接的表和全外联接的表。执行时所用左外联接、右外联接语法形式和内联接的类似。



谢谢聆听

蜗牛学院，只为成就更好的你！

