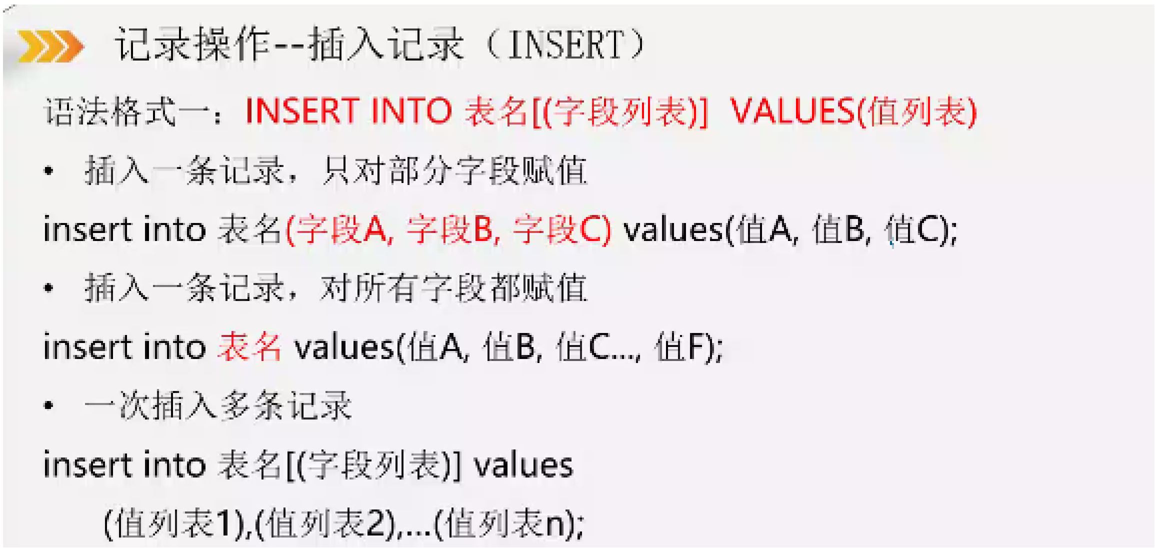
MySQL基本操作——表内操作

1、INSERT操作

```
INSERT INTO department
(department.Department_id,department.Department_name
)
VALUES
('001','数媒学院')
```



记录操作--插入记录（INSERT）

语法格式一：INSERT INTO 表名[(字段列表)] VALUES(值列表)

- 插入一条记录，只对部分字段赋值

insert into 表名(字段A, 字段B, 字段C) values(值A, 值B, 值C);

- 插入一条记录，对所有字段都赋值

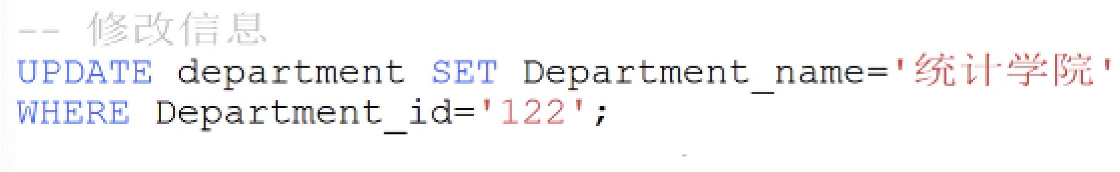
insert into 表名 values(值A, 值B, 值C..., 值F);

- 一次插入多条记录

insert into 表名[(字段列表)] values
    (值列表1),(值列表2),...(值列表n);

2、 UPDATE操作

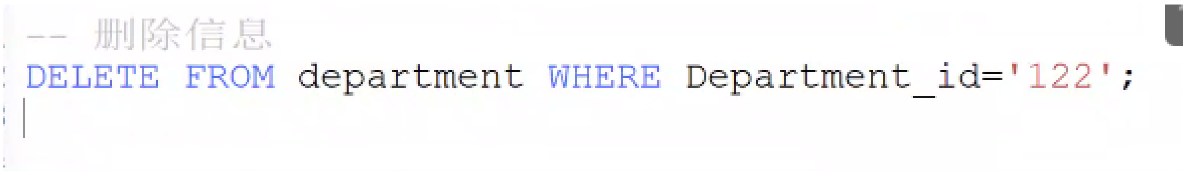
```
UPDATE department SET Department_name='信息学院'
WHERE Department_id='001';
```



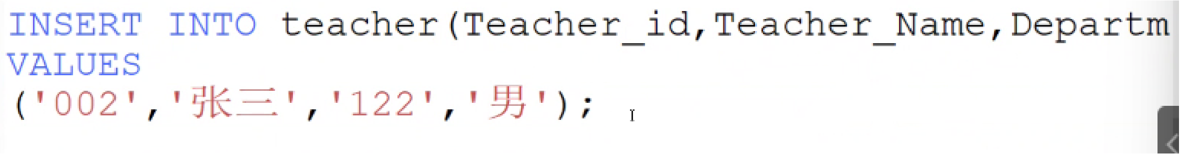
```
-- 修改信息
UPDATE department SET Department_name='统计学院'
WHERE Department_id='122';
```

3、DELETE操作

```
DELETE FROM department WHERE Department_id='001';
```



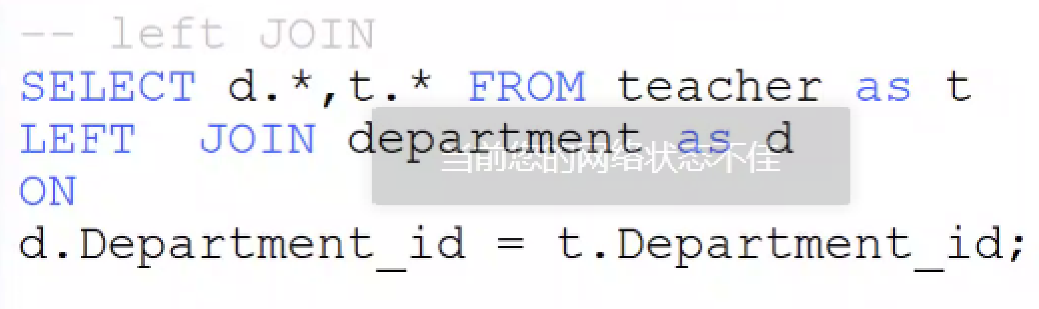
```
-- 删除信息
DELETE FROM department WHERE Department_id='122';
```



```
INSERT INTO teacher(Teacher_id,Teacher_Name,Departm
VALUES
('002','张三','122','男');
```

```
INSERT INTO teacher
(teacher.Teacher_id,teacher.Teacher_name,teacher.Department_id,teacher.Gender)
VALUES
('111','李志浩','001','男');
INSERT INTO teacher
(teacher.Teacher_id,teacher.Teacher_name,teacher.Department_id,teacher.Gender)
VALUES
('112','潘颖','002','女');
INSERT INTO teacher
(teacher.Teacher_id,teacher.Teacher_name,teacher.Department_id,teacher.Gender)
VALUES
('113','岳庆生','003','男');
```

5、左外连接



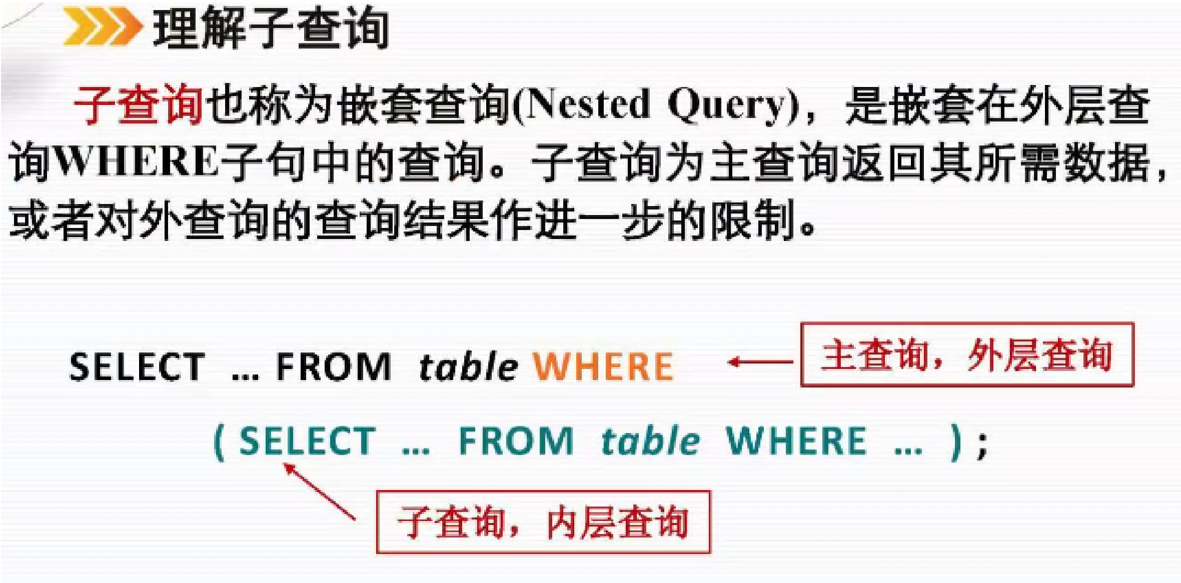
```
-- left JOIN
SELECT d.*,t.* FROM teacher as t
LEFT  JOIN department as d
ON
d.Department_id = t.Department_id;
```

当前您的网络状态不佳

```
select d.department_id from department As d;

select d.department_id from department As d
where d.department_id='001';

select * from department inner join teacher
on department.department_id = teacher.department_id;
```

```
select * from department left join teacher
on department.department_id = teacher.department_id;

select * from department right join teacher
on department.department_id = teacher.department_id;
```

6、子查询



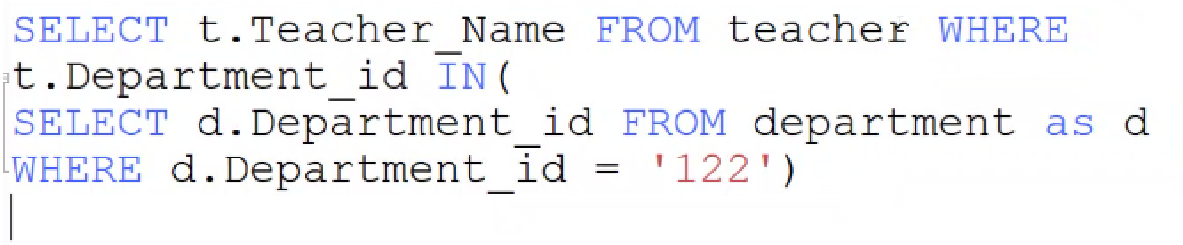
理解子查询

子查询也称为嵌套查询(Nested Query)，是嵌套在外层查询WHERE子句中的查询。子查询为主查询返回其所需数据，或者对外查询的查询结果作进一步的限制。

```
SELECT ... FROM table WHERE
    ( SELECT ... FROM table WHERE ... );
```

主查询，外层查询

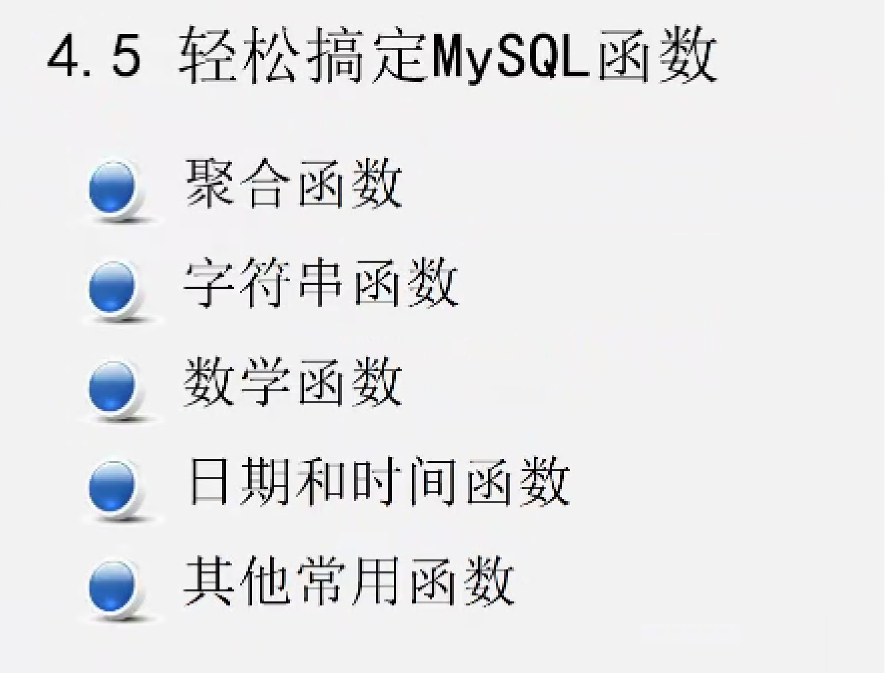
子查询，内层查询



```
SELECT t.Teacher_Name FROM teacher WHERE
t.Department_id IN(
SELECT d.Department_id FROM department as d
WHERE d.Department_id = '122')
```

```
SELECT t.Teacher_name FROM teacher as t WHERE
t.Department_id IN(
SELECT d.Department_id FROM department as d WHERE
d.Department_name = '数媒学院')
```

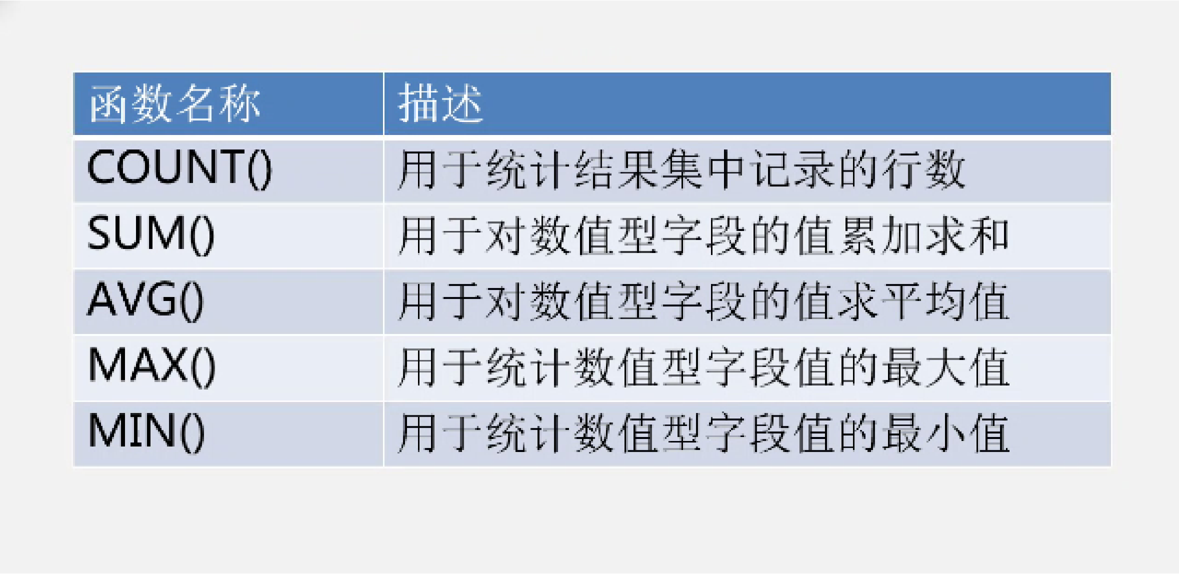
7、MySQL函数



4.5 轻松搞定MySQL函数

- 聚合函数
- 字符串函数
- 数学函数
- 日期和时间函数
- 其他常用函数

1）聚合函数

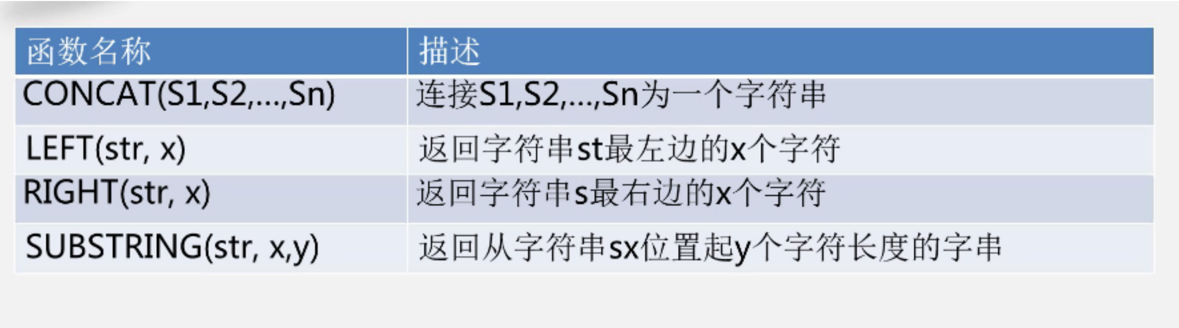


函数名称	描述
COUNT()	用于统计结果集中记录的行数
SUM()	用于对数值型字段的值累加求和
AVG()	用于对数值型字段的值求平均值
MAX()	用于统计数值型字段值的最大值
MIN()	用于统计数值型字段值的最小值

```
-- 学院数量 --
SELECT COUNT(*) as '学院数量' FROM department;
-- 求取所有成绩之和 --
```

```
SELECT SUM(c.Report) FROM choose as c
WHERE c.Report>=60;
-- 求取成绩的平均值 --
SELECT AVG(d.Report) FROM choose as d
WHERE d.Report>=0;
-- 求取成绩的最大值 --
SELECT MAX(d.Report) FROM choose as d
WHERE d.Report>=0;
-- 求取成绩的最小值 --
SELECT MIN(d.Report) FROM choose as d
WHERE d.Report;
```

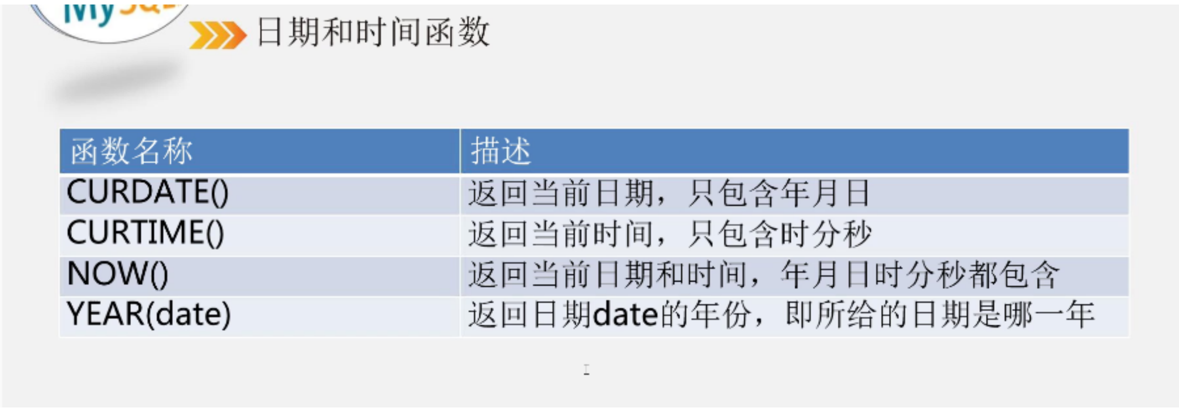
2）字符串函数



函数名称	描述
CONCAT(S1,S2,...,Sn)	连接S1,S2,...,Sn为一个字符串
LEFT(str, x)	返回字符串st最左边的x个字符
RIGHT(str, x)	返回字符串s最右边的x个字符
SUBSTRING(str, x,y)	返回从字符串sx位置起y个字符长度的字串

```
-- 连接字符 --
SELECT CONCAT(department.Department_name,'bb','cc')
FROM department;
-- 区间 --
SELECT SUBSTRING("scdvgf",3,5);
SELECT LEFT("scdvgf",4)
```

3）日期时间函数



日期和时间函数

函数名称	描述
CURDATE()	返回当前日期，只包含年月日
CURTIME()	返回当前时间，只包含时分秒
NOW()	返回当前日期和时间，年月日时分秒都包含
YEAR(date)	返回日期date的年份，即所给的日期是哪一年

```
SELECT LEFT("scdvgf",4);
-- 年月日 --
SELECT CURDATE();

-- 时分秒 --
SELECT CURTIME();

SELECT NOW();

SELECT YEAR(2022);

SELECT YEAR(NOW());
```

1、查询和'李志浩'老师在同一学院的教师的信息

1、查询哪些同学至少有一门课程的成绩为95分

1、查询哪些同学所有课程的成绩在85分以上

```
-- 1、查询和'李志浩'老师在同一学院的教师的信息 --
SELECT * FROM `teacher` WHERE teacher.Department_id
IN(
SELECT teacher.Department_id FROM teacher WHERE
teacher.Teacher_name = '李志浩'
);
-- 2、查询哪些同学至少有一门课程的成绩为95分 --
SELECT student_id,student_name FROM students
WHERE EXISTS(
  SELECT * FROM choose
  WHERE choose.student_id=students.student_id
  AND report=95);
-- 3、查询哪些同学所有课程的成绩在85分以上 --
SELECT * FROM students WHERE students.Student_id
NOT IN(
SELECT choose.Student_id FROM choose WHERE(
choose.Report<=85 OR choose.Report IS NULL)
);
-- 查询所有课程的成绩都在85分以上的同学的信息
SELECT students.* FROM students
WHERE student_id IN(
  SELECT student_id FROM choose
  GROUP BY student_id HAVING min(report)>85
);
```