

子查询：分为四种，根据返回值来划分的

子查询 返回一个值 返回一列多行值 返回一行多列值 返回 多行多列值

标量子查询：子查询的结果只返回一个值

```
-- 标量子查询
select sname from student_info where age = (select min(age) from student_info);
-- 列子查询 查询班上年龄最低的三个学生的姓名
select sname from student_info where age in
(select * from (select age from student_info order by age limit 3) as s1 );

select sname from student_info where age in (select age from student_info where age <= 20 );
-- 行子查询 包含1个以上字段 但是是一行
select * from student_info
where (sid,sname) = (select sid,sname from student_bak where age = 16)
-- 表子查询 包含多行多列
select * from student_info
where (sid,sname) in (select sid,sname from student_bak where age <= 20)
```

EXISTS 谓词:

EXISTS 是一个非常牛叉的谓词，它允许数据库高效地检查指定查询是否产生某些行

语法：只要子查询语句查询到结果，则条件成立

```
1
2 -- EXISTS 当查询结果有记录时，EXISTS 的判断为真
3 select * from student_info where EXISTS(select * from a);
4 -- 当学生存在英语考试成绩的时候，显示学生姓名
5 select sname from student_info where sid in (select sid from score_info where cid=3 )
6
```

派生表:

在子查询返回的值中，可能返回一个表，如果将子查询返回的虚拟表再次作为 FROM 子句的输入时，子查询的虚拟表就成为了一个派生表

```
-- 派生表
select s.sid,s.sname from (select * from student_info where sid <= 3) as s;
```

联合查询：在企业中，一般用来把多个表合并 或者 查出两个表直接的差异（dba 分析用的）

-- 联合查询 要求：字段数量要一致，字段值要类似包容

```
select sid,sname from student_info
UNION all -- 两个表数据合并
select sid,sname from student_bak ;
```

```
select sid,sname from student_info
UNION -- 两个表数据合并，且去除重复项
select sid,sname from student_bak ;
```

Union 可以改为 union distinct

自连接：自己连接自己

Cno (课程号)	Cname (课程名称)	Cpno (先修课程号)
1	数据库	5
2	高等数学	
3	信息系统	1
4	操作系统	6
5	数据结构	7
6	数据处理	
7	C语言	6

-- 自连接 把一个表在from中 起不同的别名, 变为多个表, 进行连接查询

```
select a.*,b.cname
from t1 as a,t1 as b
where a.cno = b.cpno;
```

视图: 是一个数据库对象, 数据库最基本的对象就是表

视图是一个虚拟表, 它由其它的查询语句组成, 视图数据来源的表称为 基表

视图语法: create view 视图名称 as 查询语句

```
create view v_a as
select sid,sname,age from student_info where sid<4;
```

视图使用: 把视图当成和表一样的数据源即可使用

```
select * from v_a where sname like '王%';
```

视图的作用: 1、开发一般操作视图, 不操作基表, 这样提高了安全性

```
create view v_a1 as
select sid as 'ssid',sname as 'sn',age as 'sa' from student_info where sid<4;

select * from v_a1 where sn like '王%';
```

2、把负责的查询语句用视图代替, 增加了数据的查询效率

视图注意事项:

1、在视图的基础上能再次创建视图

```
create view v_b as
select * from v_a where sname like '王%';
```

2、常规情况下, 修改视图的数据会关联修改基表的数据, 修改基表的数据也会修改视图的

3、视图中包含聚合函数, 不能对聚合函数的字段值进行修改

```
create view v_c as
select count(*) as 'n' from student_info;
```

```
update v_c set n = 5; -- 不能改
```

```
drop view v_a; -- 删除视图
```

索引:

相当于书籍的目录，能提供检索数据的速度 比如新华字典有目录  
在数据库中，数据也是分页存放的，一般情况下默认是 4kb/页

什么情况下书籍才会有目录，比如书的内容较多的时候才有，宣传单不会有目录  
所以，数据库中，表中字段的值取值数量越多时，建索引才合理，比如：姓名建索引、性别字段不建索引

目录会提高查询的性能，但是会降低 数据修改（增删改）的性能，比如：新华字典多了一个新汉字：1、目录更新 2、内容要更新

所以：数据库中，不能建太多的索引，索引能提高数据的查询速度，但是数据增删改速度降低，同时还会占用空间

创建索引的语法:

```
create index i_sn on student_info(sname); -- 创建索引
```

索引的使用语法:

```
select * from student_info where sname = '小明'; -- 自动使用索引
```

```
select now();
```

```
select * from student_info where sname = '小明'; -- 自动使用索引
```

```
select now();
```

```
drop index i_sn on student_info; -- 删除索引
```

备注：公司里面，索引的创建只能由 dba 决定。