## 联合查询

Union [distinct | all] 默认为distinct，将两张表中重复的记录去除

如果联合的表要排序，要将每个select语句用括号包裹起来，并且要用limit进行限量

(SELECT roleId as '第一', roleName as '第二' from t\_role)

union

(select userId, userName from t\_user)

字段的名字为第一个select中的字段名字。

## 连接查询

Select 字段列表 from 表1 [ cross | inner | left | right ] join 表2 on 连接条件

交叉连接（cross join）相当于是两张表做笛卡尔积

内连接（inner join）

左连接（left join）左外连接 主表 left outer join 从表

右连接（right join）右外连接 从表 right outer join 主表

其中，左右连接会将主表中的记录都查得出来。符合on的匹配条件的，会和其他表进行连接不符合的，也会查出来，但是其他表的值为null（表示不符合筛选条件）

如果是用了where作为筛选，那就会将不符合条件的记录去除。

先做笛卡尔积，如果有符合条件的，都保留下来，主表分裂，如果没有，主表保留，从表为null

自连接，就是利用别名，将物理上的一张表，分成逻辑上的两张表

## 子查询：先进行子查询，再进行其他查询

### 在where中的子查询

标量子查询：where 条件判断 { = | <> }

Select 字段 from 表名 [where] [ group by ] [ order by ] [ limit ]

一列多行子查询

标量子查询：where 条件判断 { in | not in }

Select 字段 from 表名 [where] [ group by ] [ order by ] [ limit ]

一行多列子查询

标量子查询：where 条件判断 =

Select 字段 from 表名 [where] [ group by ] [ order by ] [ limit ]

### 在from后面的子查询（表子查询）

再from，作为一张表，用括号将子查询括起来，同时用as设置别名

一列多行

一行一列

一行多列

多行多列

## 子查询关键字exists，any，all

Where exists( 子查询 ) 其中 exists（子查询作为个整体），如果子查询有返回结果，则整体为true，若无结果返回，则整体为false。

Any （子查询）不能作为一个整体，只能是一部分

Where 表达式 运算符 any（子查询）

Where id = any(select id from t ) 表示id只要是any集合中的一个就行，等价于 in (select )

All (子查询)

Where 表达式 运算符 all（子查询）

select \* from s where id < all(select id from t where id > 3)