



链滴

# oracle 中 union 和 union all 区别与性能分析

作者: [dianjiu](#)

原文链接: <https://ld246.com/article/1597201383049>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 一、概要

经常写sql的同学可能会用到union和union all这两个关键词,可能你知道使用它们可以将两个查询的果集进行合并,那么二者有什么区别呢?下面我们就简单的分析下

## 二、比较

union: 对两个结果集进行并集操作,不包括重复行,相当于distinct, 同时进行默认规则的排序;

union all: 对两个结果集进行并集操作,包括重复行,即所有的结果全部显示,不管是不是重复;

下面我们举一个简单的例子来证明上面的结论:

### 1、准备数据:

```
drop table student;
```

```
create table student  
(  
    id int primary key,  
    name nvarchar2(50) not null,  
    score number not null  
);
```

```
insert into student values(1,'Aaron',78);  
insert into student values(2,'Bill',76);  
insert into student values(3,'Cindy',89);  
insert into student values(4,'Damon',90);  
insert into student values(5,'Ella',73);  
insert into student values(6,'Frado',61);
```

```
insert into student values(7,'Gill',99);
insert into student values(8,'Hellen',56);
insert into student values(9,'Ivan',93);
insert into student values(10,'Jay',90);
```

```
commit;
```

## 2、比较不同点

比较一：

```
-- union all
select * from student where id < 4
union all
select * from student where id > 2 and id < 6
```

```
-- union
select * from student where id < 4
union
select * from student where id > 2 and id < 6
```

union all 查询结果:

	ID	NAME	SCORE
▶ 1	1	Aaron ...	78
2	2	Bill ...	76
3	3	Cindy ...	89
4	3	Cindy ...	89
5	4	Damon ...	90
6	5	Ella ...	73

union 查询结果:

	ID	NAME	SCORE
▶ 1	1	Aaron ...	78
2	2	Bill ...	76
3	3	Cindy ...	89
4	4	Damon ...	90
5	5	Ella ...	73

通过比较不难看出, union all不会去掉重复记录, 而union去掉了重复的记录。

比较二：

```
-- union all
select * from student where id > 2 and id < 6
union all
select * from student where id < 4
```

```
-- union
select * from student where id > 2 and id < 6
union
select * from student where id < 4
```

union all 查询结果:

	ID	NAME	SCORE
▶ 1	3	Cindy	89
2	4	Damon	90
3	5	Ella	73
4	1	Aaron	78
5	2	Bill	76
6	3	Cindy	89

union 查询结果:

	ID	NAME	SCORE
▶ 1	1	Aaron	78
2	2	Bill	76
3	3	Cindy	89
4	4	Damon	90
5	5	Ella	73

通过比较不难看出, union all会按照关联的次序组织数据, 而union会依据一定的规则进行排序.

那么这个规则是什么呢? 我们通过下面的查询得出规律:

```
-- union
select score,id,name from student where id > 2 and id < 6
union
select score,id,name from student where id < 4
```

	SCORE	ID	NAME
▶ 1	73	5	Ella
2	76	2	Bill
3	78	1	Aaron
4	89	3	Cindy
5	90	4	Damon

结论: 按照字段出现的顺序进行排序, 之前的查询相当于order by id, name, score, 刚刚的查询相当于order by score, id, name.

### 三、总结

一、因为union all仅仅是简单的合并查询结果, 并不会做去重操作, 也不会排序, 所以union all效率要union高.所以在能够确定没有重复记录的情况下, 尽量使用union all.

二、通常如果表有多个索引列时, 用union替换where子句中的or会起到较好的效果, 索引列使用or会成全表扫描.

注意: 以上规则只针对多个索引列有效, 假如有column没有被索引, 那还是用or吧.

例如: 还是使用上面的例子, 假定name和score上建有索引.

```
-- 高效
select id, name, score from student where name like '%y%'
union
select id, name, score from student where score between 80 and 90
```

```
-- 低效
select id, name, score from student where name like '%y%' or score between 80 and 90
```