1、 减少数据访问（减少磁盘访问）

创建并使用正确的索引

2、 返回更少数据（减少网络传输或磁盘访问）

分页：
1.客户端(应用程序或浏览器)分页
2.应用服务器分页
3.数据库SQL分页

oracle数据库一般采用rownum来进行分页，常用分页语法有如下两种：

直接通过rownum分页：

```
select * from (
      select a.*,rownum rn from
            (select * from product a where company_id=? order by status) a
      where rownum<=20)
where rn>10;
```

数据访问开销=索引IO+索引全部记录结果对应的表数据IO

采用rowid分页语法

```
create index myindex on product(company_id,status);
select b.* from (
      select * from (
            select a.*,rownum rn from
                  (select rowid rid,status from product a where company_id=? order by
status) a
            where rownum<=20)
      where rn>10) a, product b
where a.rid=b.rowid;
```

数据访问开销=索引IO+索引分页结果对应的表数据IO

实例：

一个公司产品有1000条记录，要分页取其中20个产品，假设访问公司索引需要50个IO，2条记录需要1个表数据IO。

那么按第一种ROWNUM分页写法，需要550 = (50+1000/2)个IO，
按第二种ROWID分页写法，只需要60=(50+20/2) 个IO;

只返回需要的字段
通过去除不必要的返回字段可以提高性能，例：
调整前：select * from product where company_id=?;
调整后：select id,name from product where company_id=?;

优点：

1、减少数据在网络上传输开销

2、减少服务器数据处理开销

3、减少客户端内存占用

4、字段变更时提前发现问题，减少程序BUG

5、如果访问的所有字段刚好在一个索引里面，则可以使用纯索引访问提高性能。

3、 减少交互次数（减少网络传输）

1. Batch DML即批处理

2.IN List-->select * from mytable where id in(:id1,id2,...,idn);
综合考虑，一般IN里面的值个数超过20个以后性能基本没什么太大变化，
也特别说明不要超过100，超过后可能会引起执行计划的不稳定性及增加数据库CPU及内存成本

3.优化业务逻辑

某移动公司推出优惠套参，活动对像为VIP会员并且2010年1，2，3月平均话费20元以上的客户。
那我们的检测逻辑为：

```
select avg(money) as avg_money from bill where phone_no='13988888888' and date between '201001' and '201003';
select vip_flag from member where phone_no='13988888888';

if avg_money>20 and vip_flag=true then

begin
 执行套参();
end;
```

如果我们修改业务逻辑为:

```
select avg(money) as  avg_money from bill where phone_no='13988888888' and date between '201001' and '201003';

if avg_money>20 then

begin
 select vip_flag from member where phone_no='13988888888';

 if vip_flag=true then

 begin
  执行套参();
 end;
end;
```

通过这样可以减少一些判断vip_flag的开销，平均话费20元以下的用户就不需要再检测是否VIP了。

如果程序员分析业务，VIP会员比例为1%，平均话费20元以上的用户比例为90%，那我们改成如下：

```
select vip_flag from member wherephone_no='13988888888';

if vip_flag=true then

begin

 select avg(money) as avg_money frombill where phone_no='13988888888' and date between '201001' and '201003';

 if avg_money>20 then

 begin

  执行套参();

 end;

end;
```

这样就只有1%的VIP会员才会做检测平均话费，最终大大减少了SQL的交互次数。

4、 减少服务器CPU开销（减少CPU及内存开销）

1. 使用绑定变量

Java中Preparestatement就是为处理绑定变量提供的对像，绑定变量有以下优点：

1、防止SQL注入

2、提高SQL可读性

3、提高SQL解析性能，不使用绑定变更我们一般称为硬解析，
使用绑定变量我们称为软解析。

2.合理使用排序

3.减少比较操作
Like模糊查询对于数据库来说不是很擅长，特别是你需要模糊检查的记录有上万条以上时，性能比较糟糕，这种情况一般可以采用专用Search或者采用全文索引方案来提高性能

5、 利用更多资源（增加资源）

1. 客户端多进程并行访问

2.数据库并行处理