> https://blog.csdn.net/qq_35036995/article/details/80298449
> https://www.cnblogs.com/liufei-yes/p/11518338.html
> https://blog.csdn.net/WYpersist/article/details/80030499

- 如果是按时间分区的表，查询时一定要使用分区限制，如果没有分区限制，会从该表的所有数据里面遍历。
- 注意sql中or的使用，or 这个逻辑必须单独括起来，否则可能引起无分区限制
- 内连接时小表放前面，大表放后面
- 只支持等值连接，不支持非等值连接
- 连接小表时使用map join 条件
- 尽量不用order by，因为order by是全局的，会只有一个reduce

hive排序

- order by：对于查询结果做全排序，只允许一个reduce处理（当数据量较大时，慎用。严格模式下，必须结合limit来使用）
- sort by：对于单个reduce的数据进行排序
- distribute by：分区排序，经常和sort by结合使用
- cluster by：相当于sort by+distribute by
- cluster by不能通过asc、desc的方式指定排序顺序，可通过distribute by column sort by column asc|desc的方式

where条件优化

由于join操作是在where操作之前执行，所以当你在执行join时，where条件并不能起到减少join数据的作用。

优化前（关系数据库不用考虑会自动优化）：

```
select m.cid,u.id from order m join customer u on m.cid =u.id where m.dt='2013-12-12';
```

优化后(where条件在map端执行而不是在reduce端执行）：

```
select m.cid,u.id from （select * from order where dt='2013-12-12'） m join customer u on m.cid =u.id;
```

group by 优化

hive.groupby.skewindata：group by操作是否允许数据倾斜，默认是false，当设置为true时，执行计划会生成两个map/reduce作业，第一个MR中会将map的结果随机分布到reduce中，达到负载均衡的目的来解决数据倾斜。

hive.groupby.mapaggr.checkinterval：map端做聚合时，group by 的key所允许的数据行数，超过该值则进行分拆，默认是100000；

设置：

hive.groupby.skewindata=true; **//如果group by过程出现倾斜应该设置为true**

set hive.groupby.mapaggr.checkinterval=100000; //这个是group的键对应的记录条数超过这个值则会进行优化，也是一个job变为两个job

count distinct优化

优化前（只有一个reduce，先去重再count负担比较大）：

```
select count(distinct id) from tablename;
```

在数据量大时，可以使用如下优化：

优化后（启动两个job，一个job负责子查询(可以有多个reduce)，另一个job负责count(1))：

```
select count(1) from (select distinct id from tablename) tmp;
select count(1) from (select id from tablename group by id) tmp;
```

优化前查询只会启动一个job来完成，完成数据去重和计数时都只在一个reduce端（即便通过mapred.reduce.tasks设置reduce的数目为多个，但实际执行时仍然只有一个）进行，优化后会启动两个job来完成，同时可以通过mapred.reduce.tasks设定更多的reduce数目，所以适合在数据量很大的情况下使用，因为初始化一个job时花费的时间也会很长。