**精益公司内部培训**

上次课程需要强调的知识点

1. select \* into 表名1 from 表名2 where 的作用是新建表结构并插入记录，还有一个作用是做更新或删除操作时做备份。
2. 如果只是复制表结构，可以使用 select \* into 表名1 from 表名2 where 1=2
3. Insert into 表名 适用于已经有表，只是插入数据的情况
4. 思考，如果sample表是一个文本文件，你如何实现简单的查找功能（index）
5. 关于事务提交的内部变量@@ERROR

**两表及多表关联查询/sql语句优化**

Sql的两表关联（以上节课sample sample2表为例）

**1、问题的提出：存在以下数据，根据数据找出对应的sample表中三个字段对应的姓名，如何实现？**

**id age class**

**1 20 9**

**3 22 11**

**5 24 13**

**7 26 15**

1. **28 17**

**Sample sample2**

习惯做法，利用for 循环，先取出id age class的值，然后到数据库查找

Select name from sample where id=1 and age=20 and class=9(习惯性写法)

2、最简单的两表关联 单个字段关联，结果是单表内容

Select \* from sample where id in (select id from sample2 )

3、单个字段关联，结果是两个表的内容

select a.\*,b.\* from sample a,sample2 b where a.id=b.id

4、两个字段关联 ，结果是两个表的内容

select a.\*,b.\* from sample a,sample2 b where a.id=b.id

And a.age=b.age

**5、三个字段关联，结果是两个表的内容**

**select a.\*,b.\* from sample a,sample2 b where a.id=b.id**

**And a.age=b.age and a.class=b.class**

**两表关联的实质，根据关联条件生成新的数据集**

1. **把查询结果当做一个表使用。**

(select a.\*,b.\* from sample a,sample2 b where a.id=b.id

And a.age=b.age and a.class=b.class) 此处有坑(列明一致导致错误)

逐步过渡

select a.id,a.age,a.class,a.name,a.sex,b.id as id1,b.age as age2 ,b.class as class2,b.name as name2,b.sex as sex2 from sample a,sample2 b where a.id=b.id

And a.age=b.age and a.class=b.class

把查询结果当做一个表来使用

select id,name,name2 from

(select a.id,a.age,a.class,a.name,a.sex,b.id as id1,b.age as id2 ,b.class as class2,b.name as name2,b.sex as sex2 from sample a,sample2 b where a.id=b.id

And a.age=b.age and a.class=b.class ) as tmp

也可以将结果插入一个临时表

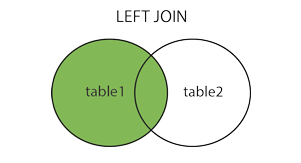
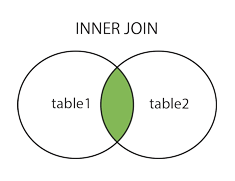
select id,name,name2 into #ls from

(select a.id,a.age,a.class,a.name,a.sex,b.id as id1,b.age as id2 ,b.class as class2,b.name as name2,b.sex as sex2 from sample a,sample2 b where a.id=b.id

And a.age=b.age and a.class=b.class ) as tmp

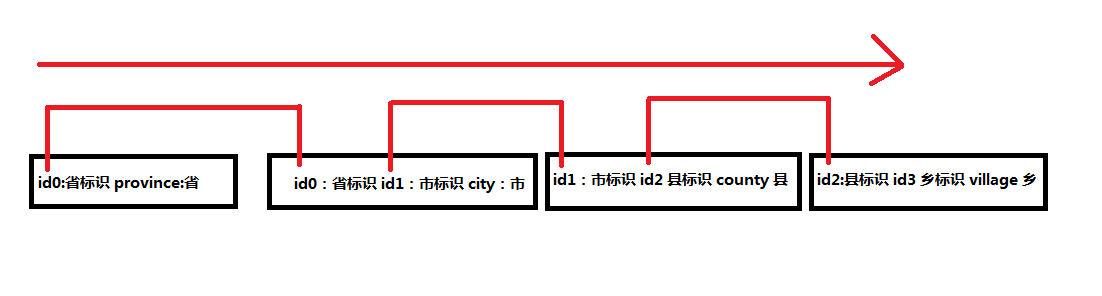
7、常用的关联类型 inner join 和left join （以上的例子属于inner join）

示意图：



inner join 取关联表的交集，left join首先确定table1的全部纪录展示，然后再关联其他表的内容，如果其他表有内容就显示，没有内容则显示null。

8、建立省市县乡模型



Province city county village

inner join 和 left join 的区别

8-1 两表关联的写法

select a.\*,b.\* from province a, city b where a.id0=b.id0

select a.\*,b.\* from province a inner join city b on a.id0=b.id0 where a.id0 =1

select a.\*,b.\* from province a left join city b on a.id0=b.id0 where a.id0 =4

8-2 三表关联的写法

select a.\*,b.\*,c.\* from province a, city b,county c where a.id0=b.id0 and b.id1=c.id1

select a.\*,b.\*,c.\* from province a

inner join city b on a.id0=b.id0

inner join county c on b.id1=c.id1

select a.\*,b.\*,c.\* from province a

left join city b on a.id0=b.id0

left join county c on b.id1=c.id1

思考以下语句有何不同

select a.\*,b.\*,c.\* from province a

left join city b on a.id0=b.id0

inner join county c on b.id1=c.id1

select a.\*,b.\*,c.\* from province a

inner join city b on a.id0=b.id0

left join county c on b.id1=c.id1

尝试将结果集作为一个表，然后进行关联的写法。

任何有查询结果的sql语句，都是从上到下，先形成结果集，然后跟下边的语句进行关联。

**Sql语句常用优化**

1、避免在索引列上使用计算．

WHERE子句中，如果索引列是函数的一部分．优化器将不使用索引而使用全表扫描．

举例:

低效：

SELECT … FROM DEPT WHERE SAL \* 12 > 25000;

高效:

SELECT … FROM DEPT WHERE SAL > 25000/12

1. GROUP BY语句的注意事项

一般在GROUP BY  的HAVING 语句之前就能剔除多余的行，所以尽量不要用它们来做剔除行的工作。他们的执行顺序应该如下最优：select 的Where字句选择所有合适的行，Group By用来分组个统计行，Having字句用来剔除多余的分组。这样Group By 个Having的开销小，查询快

1. 在IN后面值的列表中，将出现最频繁的值放在最前面，出现得最少的放在最后面，减少判断的次数
2. Like 模糊查询

有的时候会需要进行一些模糊查询比如select \* from contact where username like ‘%yue%'

关键词%yue%，由于yue前面用到了“%”，因此该查询必然走全表扫描，除非必要，否则不要在关键词前加%

5、多字段的update更新操作

同一个表的修改在一个过程里出现好几十次，如：

update table1

set col1=...

where col2=...;

update table1

set col1=...

where col2=...

......

象这类脚本其实可以很简单就整合在一个UPDATE语句来完成

6、 WHERE子句后面的条件顺序对大数据量表的查询会产生直接的影响

观察以下两个语句在执行效率上有什么不同

Select \* from 班级成绩 where 性别 = '男' and 成绩 < 60

Select \* from 班级成绩 where 成绩 < 60 and 性别 = '男'

1. 针对mysql，其条件执行顺序是 从左往右，自上而下
2. 针对orcale，其条件执行顺序是从右往左，自下而上

尽可能地第一次就过滤掉大部分无用的数据，只返回最小范围的数据

7、不要在for循环里频繁对数据库进行操作

作业

1. 分别使用inner join 和left join 关联village表（关联字段*id2*），体验不同之处
2. 将关联结果存入临时表中
3. 用inner join实现province city（关联字段*id0*） 两个表的关联，用inner join实现 county village（关联字段*id2*）两个表的关联，并将两个关联的结果作为两个表进行再次关联（关联字段*id1*）