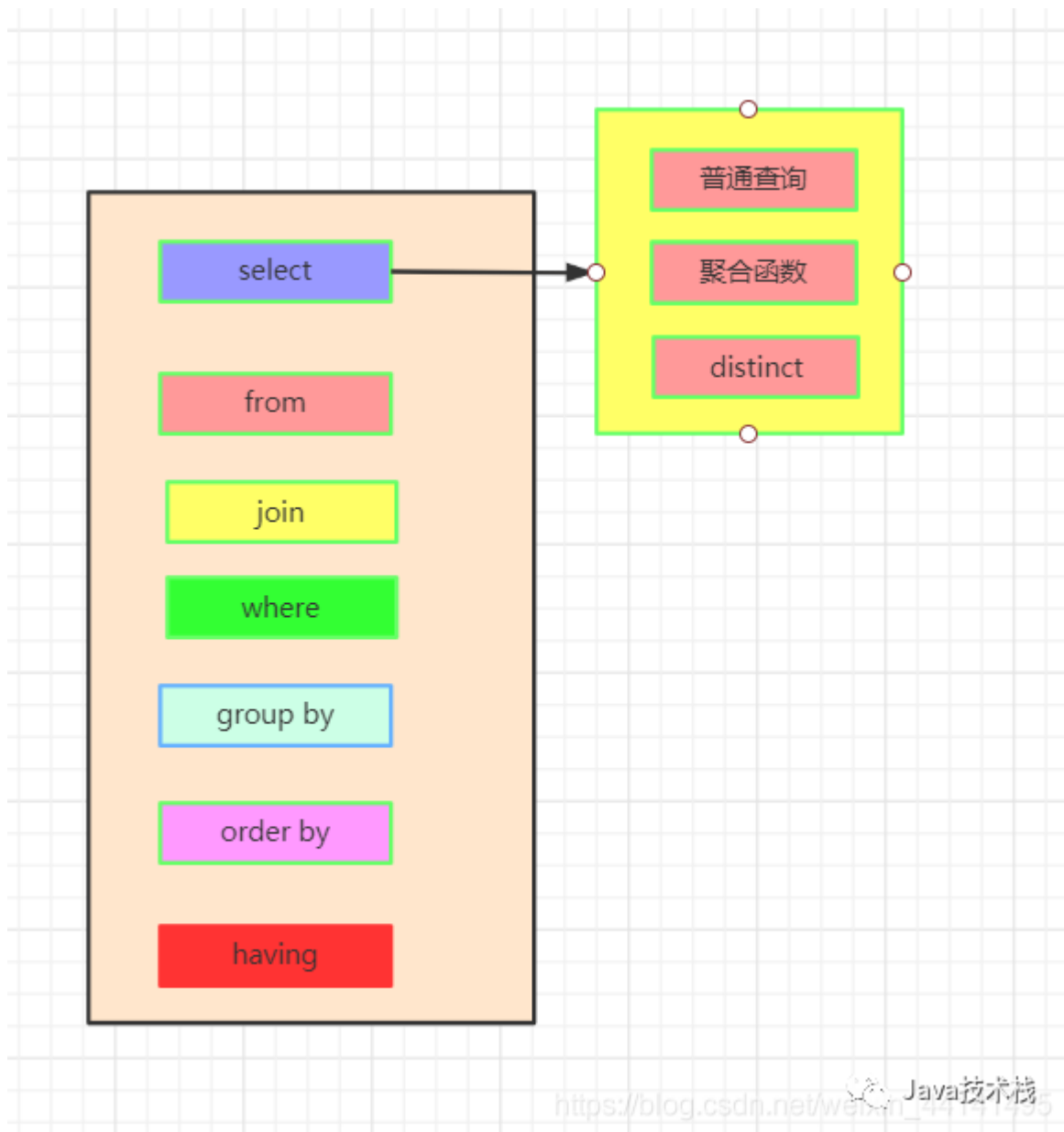


图解 SQL 执行顺序，通俗易懂！

点击关注  程序员小灰 2022-09-09 06:27 Posted on 北京

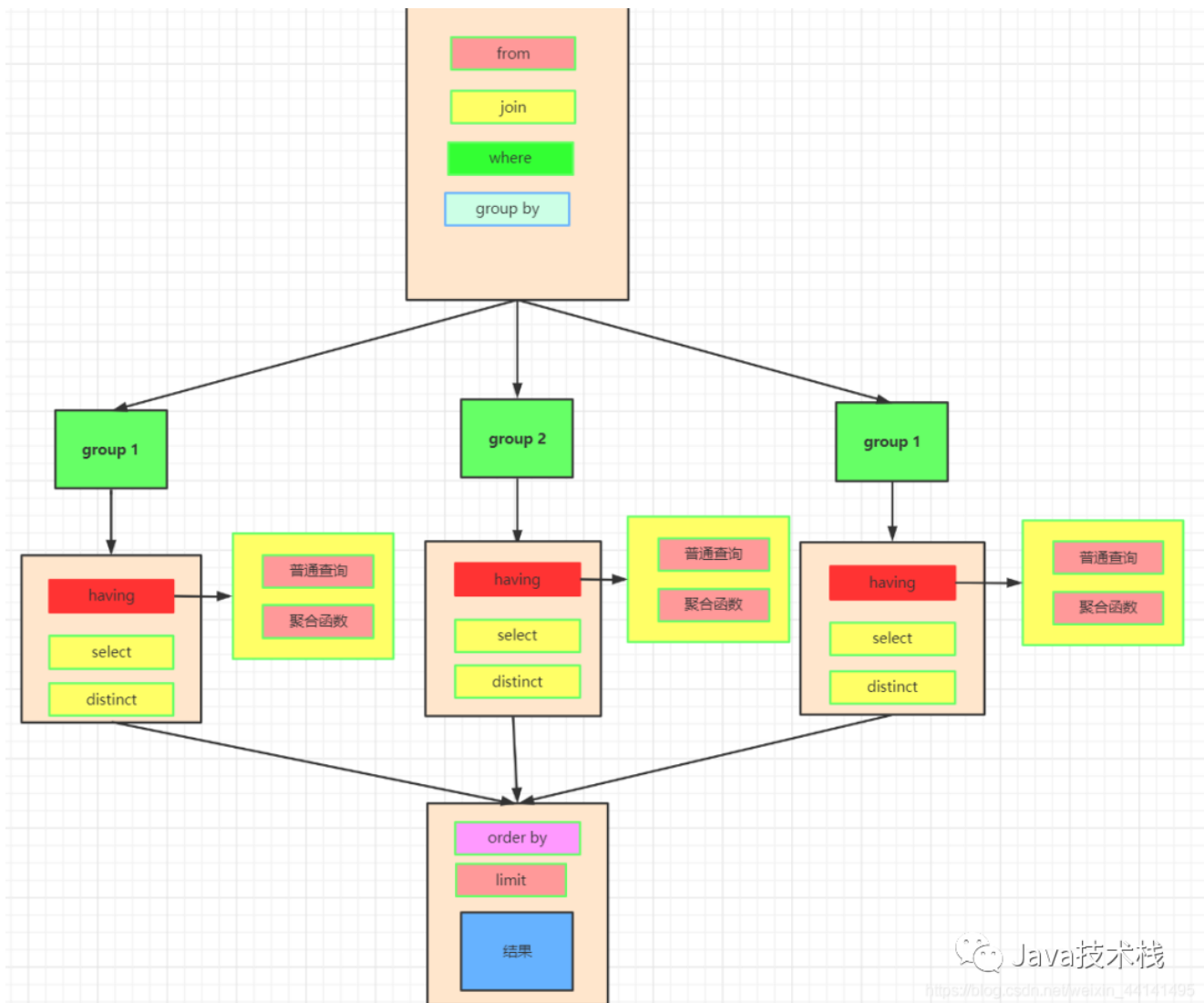
这是一条标准的查询语句:



这是我们实际上SQL执行顺序:

- 我们先执行`from`,`join`来确定表之间的连接关系，得到初步的数据
- `where`对数据进行普通的初步的筛选
- `group by` 分组
- 各组分别执行`having`中的普通筛选或者聚合函数筛选。

- 然后把再根据我们要的数据进行select，可以是普通字段查询也可以是获取聚合函数的查询结果，如果是集合函数，select的查询结果会新增一条字段
- 将查询结果去重distinct
- 最后合并各组的查询结果，按照order by的条件进行排序



数据的关联过程

数据库中的两张表

1	张三	1	8000
2	李四	2	4000
3	王五	3	5000
4	赵六	4	2000
5	张三	5	10000

6	李四	6	5000
7	王五	7	3000
8	赵六	8	20000

员工表

薪资表

Java技术栈

from&join&where

用于确定我们要查询的表的范围，涉及哪些表。另外，最新最全的 MySQL 面试题整理好了，微信搜索Java面试库小程序在线刷题。

选择一张表，然后用join连接

```
from table1 join table2 on table1.id=table2.id
```

选择多张表，用where做关联条件

```
from table1,table2 where table1.id=table2.id
```

我们会得到满足关联条件的两张表的数据，不加关联条件会出现笛卡尔积。

1	张三	1	8000
2	李四	2	4000
3	王五	3	5000
4	赵六	4	2000
5	张三	5	10000
6	李四	6	5000
7	王五	7	3000
8	赵六	8	20000

Java技术栈

group by

按照我们的分组条件，将数据进行分组，但是不会筛选数据。最新面试题整理好了，大家可以在 [Java面试库小程序在线刷题](#)。

比如我们按照id的奇偶分组

id%2=0	2	李四	2	4000
	4	赵六	4	2000
	6	李四	6	5000
	8	赵六	8	20000
id%2=1	1	张三	1	8000
	3	王五	3	5000
	5	张三	5	10000
	7	王五	7	3000

Java技术栈

having&where

having中可以是普通条件的筛选，也能是聚合函数。而where只能是普通函数，一般情况下，有having可以不写where，把where的筛选放在having里，SQL语句看上去更丝滑。

使用where再group by

先把不满足where条件的数据删除，再去分组

使用group by再having

先分组再删除不满足having条件的数据，这两种方法有区别吗，几乎没有！[点击关注公众号，Java干货及时送达](#)👉



Java技术栈

专注分享Java技术干货，包括多线程、JVM、Spring Boot、Spring Cloud、IntelliJ ID...
522篇原创内容

公众号

举个例子：

100/2=50, 此时我们把100拆分 (10+10+10+10+10...)/2=5+5+5+...+5=50 ,只要筛选条件没变, 即便是分组了也得满足筛选条件, 所以where后group by 和group by再having是不影响结果的!

不同的是, having语法支持聚合函数,其实having的意思就是针对每组的条件进行筛选。我们之前看到了普通的筛选条件是不影响的, 但是having还支持聚合函数, 这是where无法实现的。

当前数据分组情况

id%2=0	2	李四	2	4000
	4	赵六	4	2000
	6	李四	6	5000
	8	赵六	8	20000
id%2=1	1	张三	1	8000
	3	王五	3	5000
	5	张三	5	10000
	7	王五	7	3000

执行having的筛选条件, 可以使用聚合函数。筛选掉工资小于各组平均工资的 `having salary < avg(salary)`

8	赵六	8	20000
1	张三	1	8000
5	张三	5	10000

select

分组结束之后，我们再执行select语句，因为聚合函数是依赖于分组的，聚合函数会单独新增一个查询出来的字段，这里用紫色表示，这里我们两个id重复了，我们就保留一个id，重复字段名需要指向来自哪张表，否则会出现唯一性问题。最后按照用户名去重。

```
select employee.id,distinct name,salary, avg(salary)
```

8	赵六	20000	7750
1	张三	8000	6500

Java技术栈

将各组having之后的数据再合并数据。最新面试题整理好了，大家可以在 [Java面试库小程序](#) 在线刷题。

8	赵六	20000	7750
1	张三	8000	6500

Java技术栈

order by

最后我们执行order by 将数据按照一定顺序排序，比如这里按照id排序。如果此时有limit那么查询到相应的我们需要的记录数时，就不继续往下查了。

1	张三	8000	6500
8	赵六	20000	7750

Java技术栈

limit

记住limit是最后查询的，为什么呢？假如我们要查询年级最小的三个数据，如果在排序之前就截取到3个数据。实际上查询出来的不是最小的三个数据而是前三个数据了，记住这一点。

我们如果limit 0,3窃取前三个数据再排序，实际上最少工资的是2000,3000,4000。你这里只能是4000,5000,8000了。



1	张三	1	8000
2	李四	2	4000
3	王五	3	5000
4	赵六	4	2000
5	张三	5	10000
6	李四	6	5000
7	王五	7	3000
8	赵六	8	20000

Java技术栈

原文链接：https://blog.csdn.net/weixin_44141495/article/details/108744720

版权声明：本文为CSDN博主「程序员小章」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

End



程序员小灰

一群喜爱编程技术和算法的小仓鼠。

444篇原创内容

公众号

People who liked this content also liked