

## 03 | 学会用数据库的方式思考SQL是如何执行的

2019-06-17 陈旻



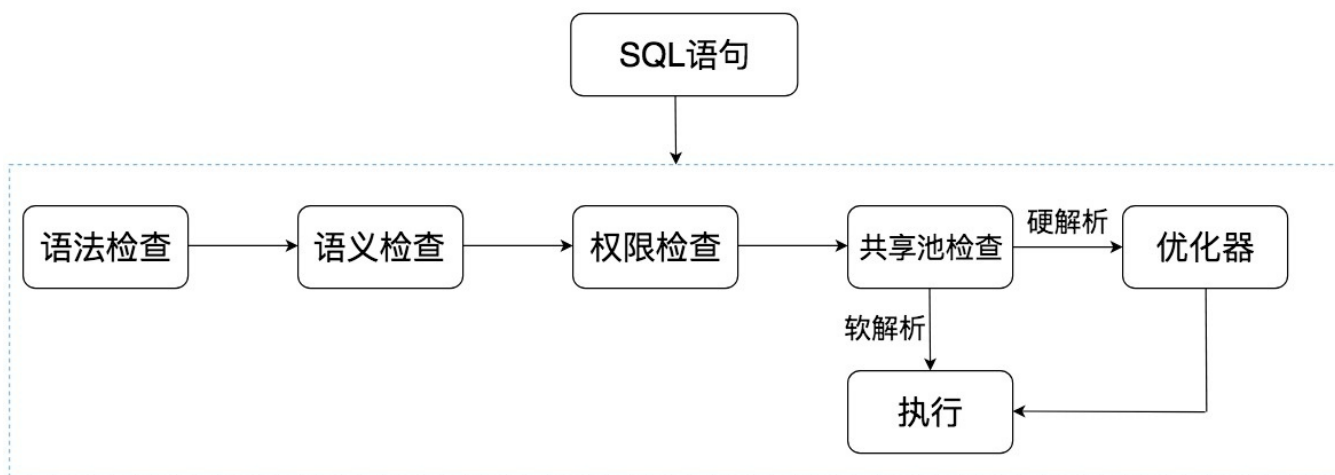
通过上一篇文章对不同的DBMS的介绍，你应该对它们有了一些基础的了解。虽然SQL是声明式语言，我们可以像使用英语一样使用它，不过在RDBMS（关系型数据库管理系统）中，SQL的实现方式还是有差别的。今天我们就从数据库的角度来思考一下SQL是如何被执行的。

关于今天的内容，你会从以下几个方面进行学习：

1. Oracle中的SQL是如何执行的，什么是硬解析和软解析；
2. MySQL中的SQL是如何执行的，MySQL的体系结构又是怎样的；
3. 什么是存储引擎，MySQL的存储引擎都有哪些？

### Oracle中的SQL是如何执行的

我们先来看下SQL在Oracle中的执行过程：



从上面这张图中可以看出，SQL语句在Oracle中经历了以下的几个步骤。

1. 语法检查：检查SQL拼写是否正确，如果不正确，Oracle会报语法错误。
2. 语义检查：检查SQL中的访问对象是否存在。比如我们在写SELECT语句的时候，列名写错了，系统就会提示错误。语法检查和语义检查的作用是保证SQL语句没有错误。
3. 权限检查：看用户是否具备访问该数据的权限。
4. 共享池检查：共享池（Shared Pool）是一块内存池，最主要的作用是缓存SQL语句和该语句的执行计划。Oracle通过检查共享池是否存在SQL语句的执行计划，来判断进行软解析，还是硬解析。那软解析和硬解析又该怎么理解呢？

在共享池中，Oracle首先对SQL语句进行Hash运算，然后根据Hash值在库缓存（Library Cache）中查找，如果存在SQL语句的执行计划，就直接拿来执行，直接进入“执行器”的环节，这就是软解析。

如果没有找到SQL语句和执行计划，Oracle就需要创建解析树进行解析，生成执行计划，进入“优化器”这个步骤，这就是硬解析。

5. 优化器：优化器中就是要进行硬解析，也就是决定怎么做，比如创建解析树，生成执行计划。
6. 执行器：当有了解析树和执行计划之后，就知道了SQL该怎么被执行，这样就可以在执行器中执行语句了。

共享池是Oracle中的术语，包括了库缓存，数据字典缓冲区等。我们上面已经讲到了库缓存区，它主要缓存SQL语句和执行计划。而数据字典缓冲区存储的是Oracle中的对象定义，比如表、视图、索引等对象。当对SQL语句进行解析的时候，如果需要相关的数据，会从数据字典缓冲区中提取。

库缓存这一个步骤，决定了SQL语句是否需要进行硬解析。为了提升SQL的执行效率，我们应该

尽量避免硬解析，因为在**SQL**的执行过程中，创建解析树，生成执行计划是很消耗资源的。

你可能会问，如何避免硬解析，尽量使用软解析呢？在**Oracle**中，绑定变量是它的一大特色。绑定变量就是在**SQL**语句中使用变量，通过不同的变量取值来改变**SQL**的执行结果。这样做的好处是能提升软解析的可能性，不足之处在于可能会导致生成的执行计划不够优化，因此是否需要绑定变量还需要视情况而定。

举个例子，我们可以使用下面的查询语句：

```
SQL> select * from player where player_id = 10001;
```

你也可以使用绑定变量，如：

```
SQL> select * from player where player_id = :player_id;
```

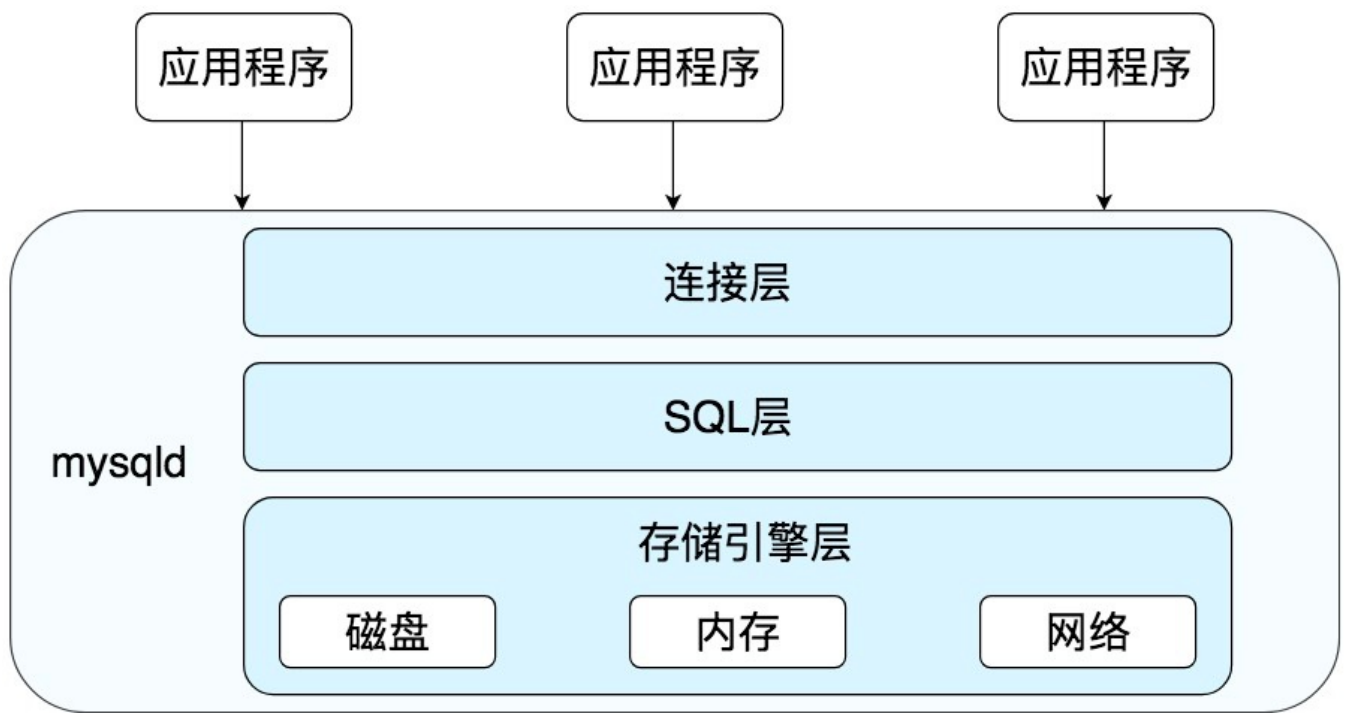
这两个查询语句的效率在**Oracle**中是完全不同的。如果你在查询`player_id = 10001`之后，还会查询10002、10003之类的数据，那么每一次查询都会创建一个新的查询解析。而第二种方式使用了绑定变量，那么在第一次查询之后，在共享池中就会存在这类查询的执行计划，也就是软解析。

因此我们可以通过使用绑定变量来减少硬解析，减少**Oracle**的解析工作量。但是这种方式也有缺点，使用动态**SQL**的方式，因为参数不同，会导致**SQL**的执行效率不同，同时**SQL**优化也会比较困难。

## MySQL中的SQL是如何执行的

**Oracle**中采用了共享池来判断**SQL**语句是否存在缓存和执行计划，通过这一步骤我们可以知道应该采用硬解析还是软解析。那么在**MySQL**中，**SQL**是如何被执行的呢？

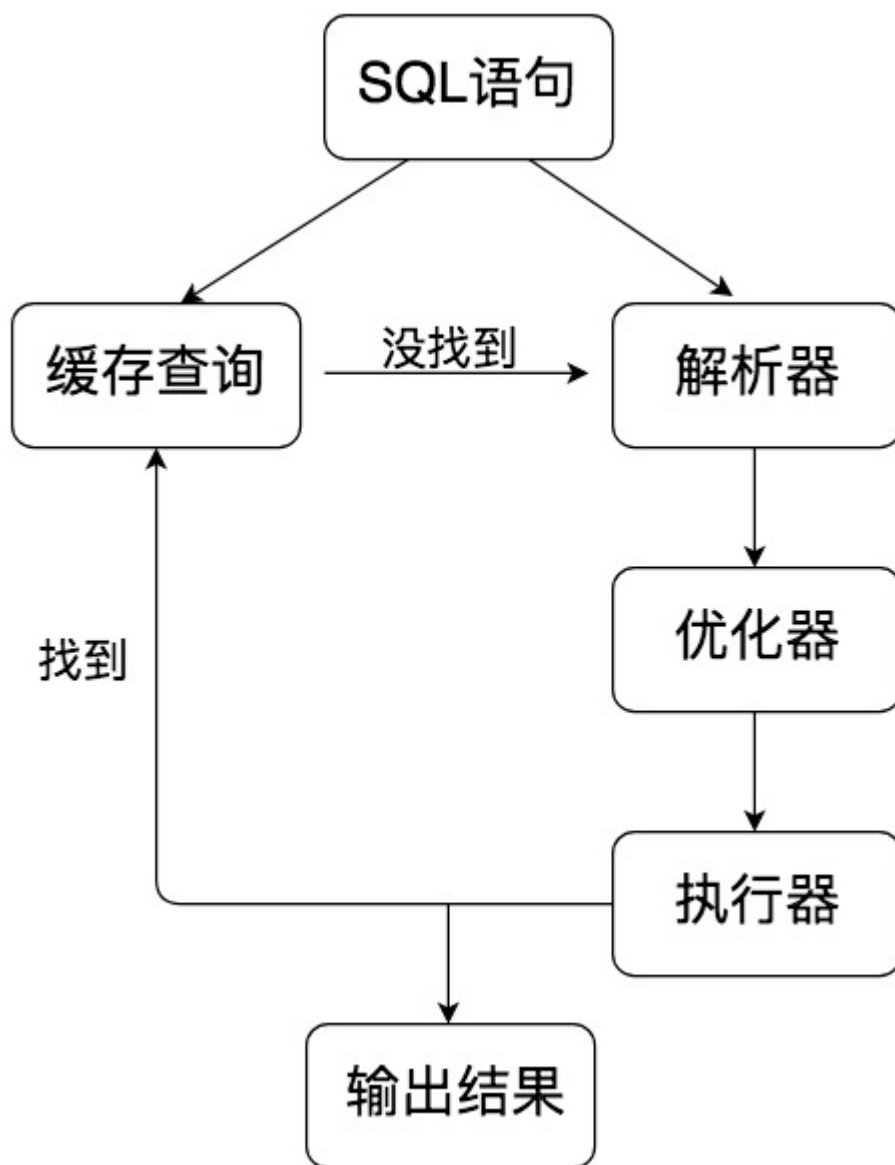
首先**MySQL**是典型的C/S架构，即Client/Server架构，服务器端程序使用的mysqld。整体的**MySQL**流程如下图所示：



你能看到MySQL由三层组成：

1. 连接层：客户端和服务端建立连接，客户端发送SQL至服务端；
2. SQL层：对SQL语句进行查询处理；
3. 存储引擎层：与数据库文件打交道，负责数据的存储和读取。

其中SQL层与数据库文件的存储方式无关，我们来看下SQL层的结构：



1. 查询缓存：Server如果在查询缓存中发现了这条SQL语句，就会直接将结果返回给客户端；如果没有，就进入到解析器阶段。需要说明的是，因为查询缓存往往效率不高，所以在MySQL8.0之后就抛弃了这个功能。
2. 解析器：在解析器中对SQL语句进行语法分析、语义分析。
3. 优化器：在优化器中会确定SQL语句的执行路径，比如是根据全表检索，还是根据索引来检索等。
4. 执行器：在执行之前需要判断该用户是否具备权限，如果具备权限就执行SQL查询并返回结果。在MySQL8.0以下的版本，如果设置了查询缓存，这时会将查询结果进行缓存。

你能看到SQL语句在MySQL中的流程是：SQL语句→缓存查询→解析器→优化器→执行器。在一部分中，MySQL和Oracle执行SQL的原理是一样的。

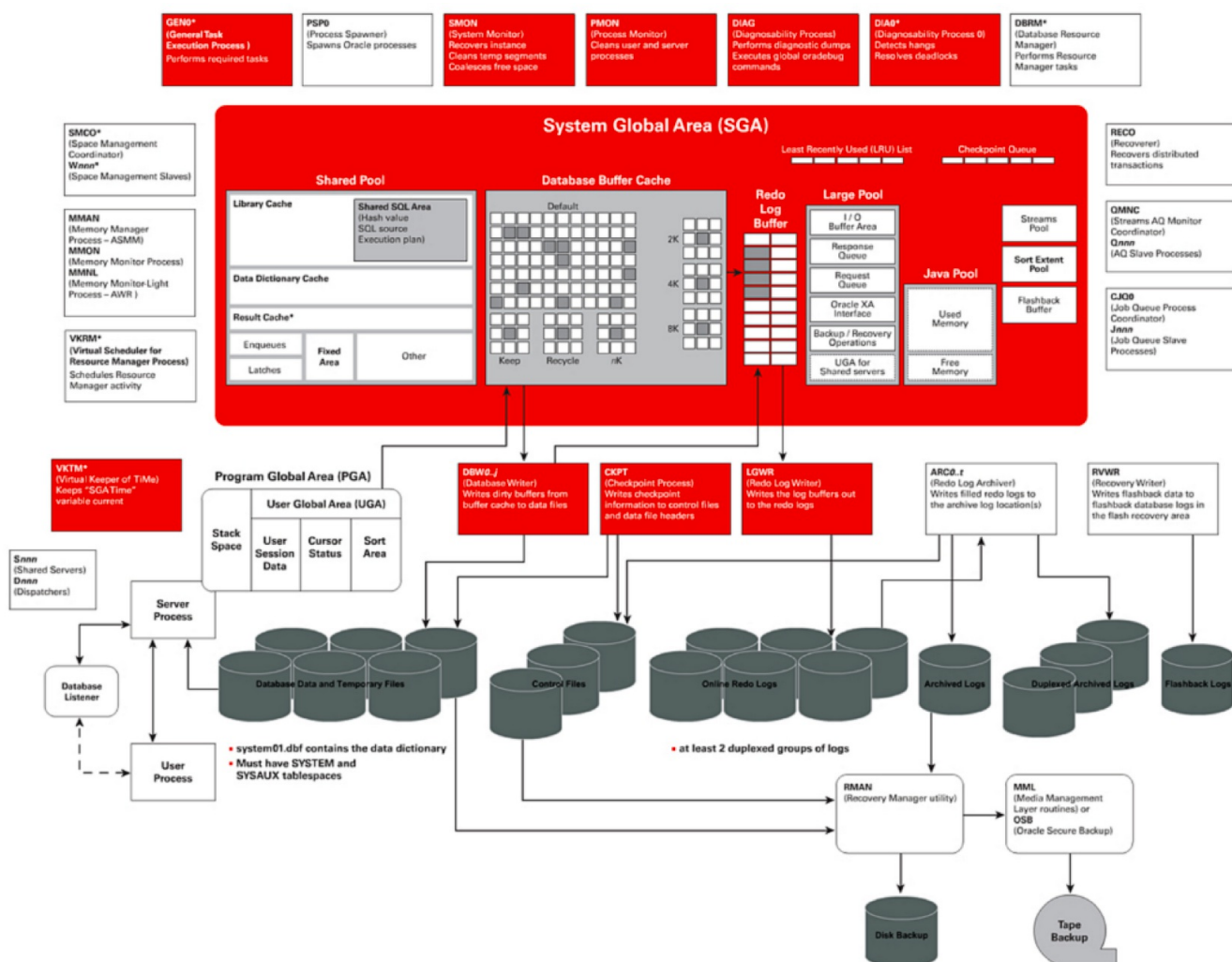
与Oracle不同的是，MySQL的存储引擎采用了插件的形式，每个存储引擎都面向一种特定的数据库应用环境。同时开源的MySQL还允许开发人员设置自己的存储引擎，下面是一些常见的存储引擎：

1. **InnoDB**存储引擎：它是MySQL 5.5版本之后默认的存储引擎，最大的特点是支持事务、行级锁定、外键约束等。
2. **MyISAM**存储引擎：在MySQL 5.5版本之前是默认的存储引擎，不支持事务，也不支持外键，最大的特点是速度快，占用资源少。
3. **Memory**存储引擎：使用系统内存作为存储介质，以便得到更快的响应速度。不过如果mysqld进程崩溃，则会导致所有的数据丢失，因此我们只有当数据是临时的情况下才使用Memory存储引擎。
4. **NDB**存储引擎：也叫做NDB Cluster存储引擎，主要用于MySQL Cluster分布式集群环境，类似于Oracle的RAC集群。
5. **Archive**存储引擎：它有很好的压缩机制，用于文件归档，在请求写入时会进行压缩，所以也经常用来做仓库。

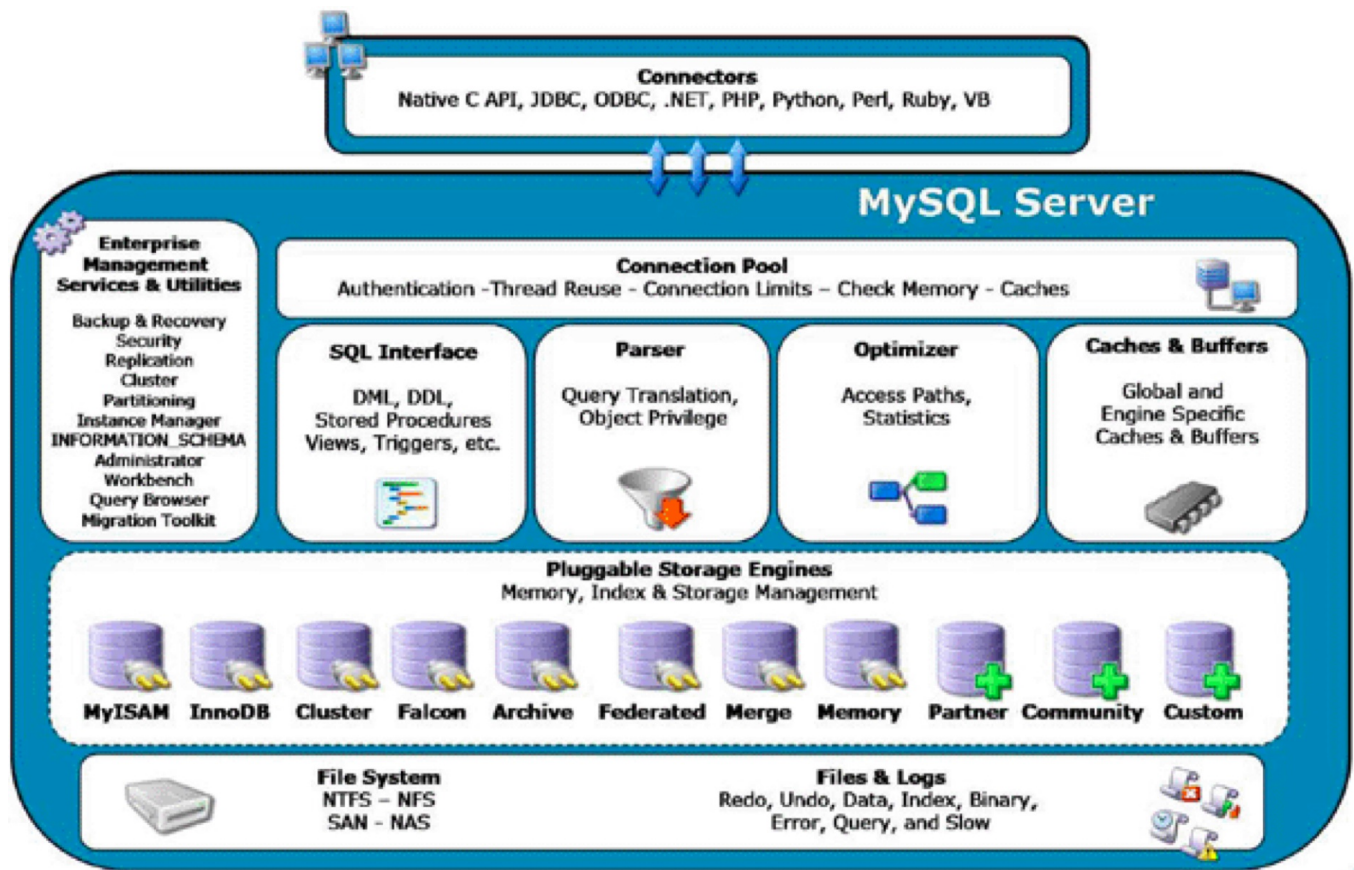
需要注意的是，数据库的设计在于表的设计，而在MySQL中每个表的设计都可以采用不同的存储引擎，我们可以根据实际的数据处理需要来选择存储引擎，这也是MySQL的强大之处。

## 数据库管理系统也是一种软件

我们刚才了解了SQL语句在Oracle和MySQL中的执行流程，实际上完整的Oracle和MySQL结构图要复杂得多：







如果你只是简单地把MySQL和Oracle看成数据库管理系统软件，从外部看难免会觉得“晦涩难懂”，毕竟组织结构太多了。我们在学习的时候，还需要具备抽象的能力，抓取最核心的部分：**SQL**的执行原理。因为不同的DBMS的SQL的执行原理是相通的，只是在不同的软件中，各有各的实现路径。

既然一条SQL语句会经历不同的模块，那我们来看一下，在不同的模块中，SQL执行所使用的资源（时间）是怎样的。下面我来教你如何在MySQL中对一条SQL语句的执行时间进行分析。

首先我们需要看下profiling是否开启，开启它可以让MySQL收集在SQL执行时所使用的资源情况，命令如下：

```
mysql> select @@profiling;
```

```
mysql> select @@profiling;
+-----+
| @@profiling |
+-----+
|           0 |
+-----+
1 row in set, 1 warning (0.00 sec)
```

profiling=0代表关闭，我们需要把profiling打开，即设置为1：

```
mysql> set profiling=1;
```

然后我们执行一个SQL查询（你可以执行任何一个SQL查询）：

```
mysql> select * from wucai.heros;
```

查看当前会话所产生的所有profiles：

```
mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
|         1 | 0.00064700 | select @@profiling |
|         2 | 0.00619700 | select * from wucai.heros |
+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

你会发现我们刚才执行了两次查询，QueryID分别为1和2。如果我们想要获取上一次查询的执行时间，可以使用：

```
mysql> show profile;
```



```
mysql> show profile;
```

Status	Duration	
starting	0.000081	
checking permissions	0.000010	权限检查
Opening tables	0.004512	打开表
init	0.000020	初始化
System lock	0.000016	锁系统
optimizing	0.000010	优化查询
statistics	0.000026	
preparing	0.000025	准备
executing	0.000005	执行
Sending data	0.001185	
end	0.000021	
query end	0.000007	
waiting for handler commit	0.000012	
query end	0.000010	
closing tables	0.000013	
freeing items	0.000220	
cleaning up	0.000026	

17 rows in set, 1 warning (0.00 sec)

当然你也可以查询指定的Query ID，比如：

```
mysql> show profile for query 2;
```

查询SQL的执行时间结果和上面是一样的。

在8.0版本之后，MySQL不再支持缓存的查询，原因我在上文已经说过。一旦数据表有更新，缓存都将清空，因此只有数据表是静态的时候，或者数据表很少发生变化时，使用缓存查询才有价值，否则如果数据表经常更新，反而增加了SQL的查询时间。

你可以使用select version()来查看MySQL的版本情况。

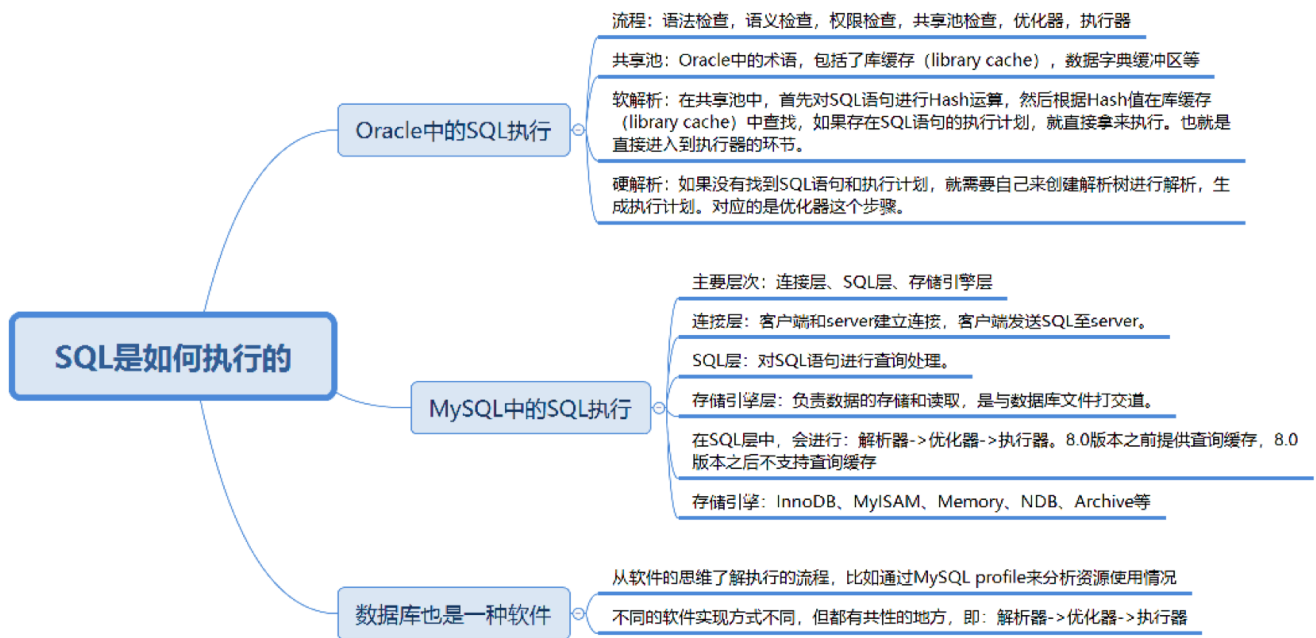
```
mysql> select version();
+-----+
| version() |
+-----+
| 8.0.13    |
+-----+
1 row in set (0.01 sec)
```

## 总结

我们在使用SQL的时候，往往只见树木，不见森林，不会注意到它在各种数据库软件中是如何执行的，今天我们从全貌的角度来理解这个问题。你能看到不同的RDBMS之间有相同的地方，也有不同的地方。

相同的地方在于Oracle和MySQL都是通过解析器→优化器→执行器这样的流程来执行SQL的。

但Oracle和MySQL在进行SQL的查询上面有软件实现层面的差异。Oracle提出了共享池的概念，通过共享池来判断是进行软解析，还是硬解析。而在MySQL中，8.0以后的版本不再支持查询缓存，而是直接执行解析器→优化器→执行器的流程，这一点从MySQL中的show profile里也能看到。同时MySQL的一大特色就是提供了各种存储引擎以供选择，不同的存储引擎有各自的使用场景，我们可以针对每张表选择适合的存储引擎。



今天的内容到这里就结束了，你能说一下Oracle中的绑定变量是什么吗？使用它有什么优缺点吗？MySQL的存储引擎是一大特色，其中MyISAM和InnoDB都是常用的存储引擎，这两个搜索引擎的特性和使用场景分别是什么？

最后留一道选择题吧，解析后的SQL语句在Oracle的哪个区域中进行缓存？

- A. 数据缓冲区
- B. 日志缓冲区
- C. 共享池
- D. 大池

欢迎你在评论区写下你的思考，我会在评论区与你一起交流，如果这篇文章帮你理顺了Oracle和MySQL执行SQL的过程，欢迎你把它分享给你的朋友或者同事。

※注：本篇文章出现的图片请点击[这里](#)下载高清图。

# SQL 必知必会

## 从入门到数据实战

陈旻

清华大学计算机博士



新版升级：点击「👤请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

### 精选留言



we

👍 0

老师，那两张，**oracle**，**mysql** 的大图。是哪儿的。有没有高清的啊。很多小字看不清楚。能否给个高清的链接。

2019-06-19

编辑回复

请您点击这里进行下载：<https://github.com/cystanford/SQL-XMind>

2019-06-19



FATMAN89

👍 0

老师讲的挺好的，想请问老师，课程所用到的数据库在哪里可以获得呢，多谢

2019-06-18

编辑回复

第五篇文章会给出下载链接～

2019-06-18



KaitoShy

👍 24

1. 绑定变量概念：**sql**语句中使用变量，通过不同的变量值来改变**sql**的执行结果

优点：减少硬解析，减少**Oracle**的工作量

缺点：参数不同导致执行效率不同，优化比较难做。

2. **MyISAM**的使用场景为读写分离的读库，而**InnoDB**为写库

3. **C**共享池，看图中有个**Shared SQL Area**。

2019-06-17



墨禾

16

### 【回答3个问题】

1、oracle中的绑定变量指的是 sql语句在执行时，通过改变不同的变量值来改变sql的执行结果；

优点：避免硬解析，提高SQL语句执行的效率；

缺点：如果使用绑定变量，那么优化器就会忽略直方图的信息，在生成执行计划的时候可能不够优化。

### 2、MyISAM:

不支持事务、外键，速度快、资源占有少；

InnoDB: 支持事务、外键、聚集索引，5.6版本以后的mysql支持全文索引；

使用场景：

需要支持事物的场景考虑InnoDB；

以读为主的数据表用MyISAM；

MyISAM奔溃后的系统的恢复较困难，没有要求的话可以用；

不知道选什么数据库合适的话，用InnoDB不会差【5.5版本以后的mysql默认的引擎是InnoDB】

### 3、oracle在共享池中进行缓存。

### 【学习总结】

#### 1、get一个学习方法

培养抽象事物的能力，掌握学习要点，如sql知识，重点掌握sql执行的原理，因为在不同的数据库中，sql执行的原理大同小异，只是在执行的顺序上有所不同。

#### 2、get一个oracle sql优化的技巧

通过绑定变量，优化sql结果

#### 3、get一个oracle、mysql共同的执行sql的原理

SQL语句-》缓存查询-》解析器-》优化器-》执行器

【PS:老师后面能不能推荐一些实战项目练习，来巩固知识点呢？】

2019-06-17



Zain Lau

15

今年考研必上北京邮电大学！

2019-06-18



虫子的一天

11

老师好，我原来用SQLSERVER比较多，经常会碰到参数化的SQL查询中，因为SQLSERVER已经缓存了查询计划，导致某些特定参数查询效率很低的事情(刚才文中也有提及Oracle也有类似问题)。

我刚听讲似乎MySQL是没这个机制的，是否MySQL就不会碰到类似问题？

另外如果不让SQLserver使用缓存的查询计划，每次都重新生成，又导致CPU高，MYSQL又是

如何避免类似问题的  
感谢

2019-06-17



allean  
共享池

👍 4

2019-06-17



跳跳

👍 3

绑定变量：在SQL语句中使用变量，通过不同的变量取值来改变SQL的执行结果  
优点是可以做软解析，避免创建解析树，生成执行计划，提高oracle的运算效率  
缺点是可能会导致执行计划不够优化

InnoDB 存储引擎：最大特点是支持事务、行级锁定、外键约束等

MyISAM：不支持事务、也不支持外键，最大特点是速度快、占用资源少

选择题选C，共享池的主要作用是缓存SQL语句和该语句的执行计划

2019-06-17



Nixus

👍 2

老师,我想提个建议,可以吗?

MySQL 的存储引擎是一大特色,其中 MyISAM 和 InnoDB 都是常用的存储引擎,这两个搜索引擎的特性和使用...

这里好像写错了,前面是存储引擎,后面变成搜索引擎了;

另外还有三个疑问,希望老师可以答疑:

1. <<MySQL实战45讲>>中,林老师讲到的是,MySQL可以分为两层: Server层和存储引擎层(连接器是属于Server层),您这里讲的是,分三层: 连接层, SQL层 和 存储引擎层,这个分层是没有严格的定义的吗?

2. 上面那个图, <https://static001.geekbang.org/resource/image/d6/a0/d6ab4e24003df69503bb055810a29fa0.jpg>

缓存查询 这一过程结束后,在最后 有两个箭头,一个指向了 执行器,另一个指向了 输出结果,这个略感迷惑;

3. InnoDB是在5.5.5作为默认存储引擎的,还是在5.5.8作为默认存储引擎的呢? 查了MySQL的网站, <https://dev.mysql.com/doc/relnotes/mysql/5.5/en/news-5-5-5.html#mysqld-5-5-5-innodb> 好像是5.5.5?

感谢老师的课程! 谢谢!

收获很大!

2019-06-19



DemonLee

👍 2

另一篇文章说: 5.5.5以上默认Engine是Innodb, 这篇说是5.5.8, 所以谁是对的?

2019-06-17



Sam

👍 2



## 1.什么是绑定变量

Oracle的绑定变量即是把谓词条件的具体值更换成一个变量，这样Oracle在共享池做hash映射时，同一类SQL但谓词条件值不同的，可以共享执行计划，进行软解析而不需要硬解析，例如  
`select * from tab where id=1`和`select * from tab where id=10`改成绑定变量  
`select * from tab where id=:a;`

## 2.绑定变量优缺点

优点：减少硬解析，加快SQL处理过程，减少系统开销；

缺点：绑定变量对应到不同的值，由于数据分布的问题，可能最佳的执行计划不一定是同一个，但是绑定变量在软解析时就固定对应到特定的执行计划上。

解析后的SQL语句存放在Oracle的共享池。

想请教老师一个问题，MySQL没有软硬解析的设计，那MySQL是怎么应对硬解析的低效呢？

2019-06-17



redrain

👍 2

老师，类似状态，**status**这种有几个固定值的列，有必要建立索引吗，广泛流传不需要建立，但是我发现建立有提升查询效率，**innodb**引擎5.7

2019-06-17



DriveMan\_邱佳源

👍 1

老师，我想问一个问题就是mysql的用户权限检查，为什么是在优化器优化之后检查的，如果用户没有权限，那优化这一步不就是浪费更多资源吗，为什么不是在解析器解析之后执行权限检查的。

2019-06-30



To you

👍 1

@吴青 回答下这位老哥的问题，在sql执行的时候 可能有触发器 得在执行器阶段才能确定的

2019-06-29



吴青

👍 1

老师，mysql的sql层的鉴权为什么要放在第四步，语义分析完后是不是就可以鉴权了，省去了生成执行路径的开销。mysql为什么这么做呢

2019-06-21



ABC

👍 1

1. 绑定的变量是为了SQL语句能被缓存，提高查询效率，不足是会让SQL优化变得困难；
2. MyASIM不支持事务，不支持外键，但速度很快，它是5.5.8之前的默认存储引擎，InnoDB支持事务，支持行级锁，它是5.5.8之后的默认引擎。

3.解析后的SQL存在Oracle的共享池中。

我有个疑问，MySQL的查询缓存是直接缓存的查询结果吗？还是跟Oracle一样缓存的SQL语句呢？谢谢老师。

2019-06-17



一步

👍 1

对于mysql存储过程也是可以传递参数，执行相同的sql，那这个和oracle的软解析原理是一样吗？如果和软解析一样效率会提升，为什么在mysql中大家都不去用存储过程呢？希望老师解答一下

2019-06-17



放~飞de心~空

👍 1

Sql Server 绑定变量是怎样的？只会出现现在存储过程中吗？比如这样：

```
ALTER Procedure [dbo].[SO_SA_census2](  
    @os_dd varchar(50),  
    @sort int,  
    --@dep varchar(50),  
    --@sal_no varchar(50),  
    @wh varchar(50),  
    @prd_no varchar(50)  
    ) As
```

2019-06-17

这个需求  
做不了

啦啦啦

👍 1

老师，oracle的共享池会缓存sql语句的执行结果吗

2019-06-17



Regis

👍 1

补充一下KaitoShy的，利用绑定变量，影响执行效率的原因，是由于每个表里面的使用绑定变量的列的数据分布不平均导致，如果列值分布差别不大的话就会避免这个问题

2019-06-17