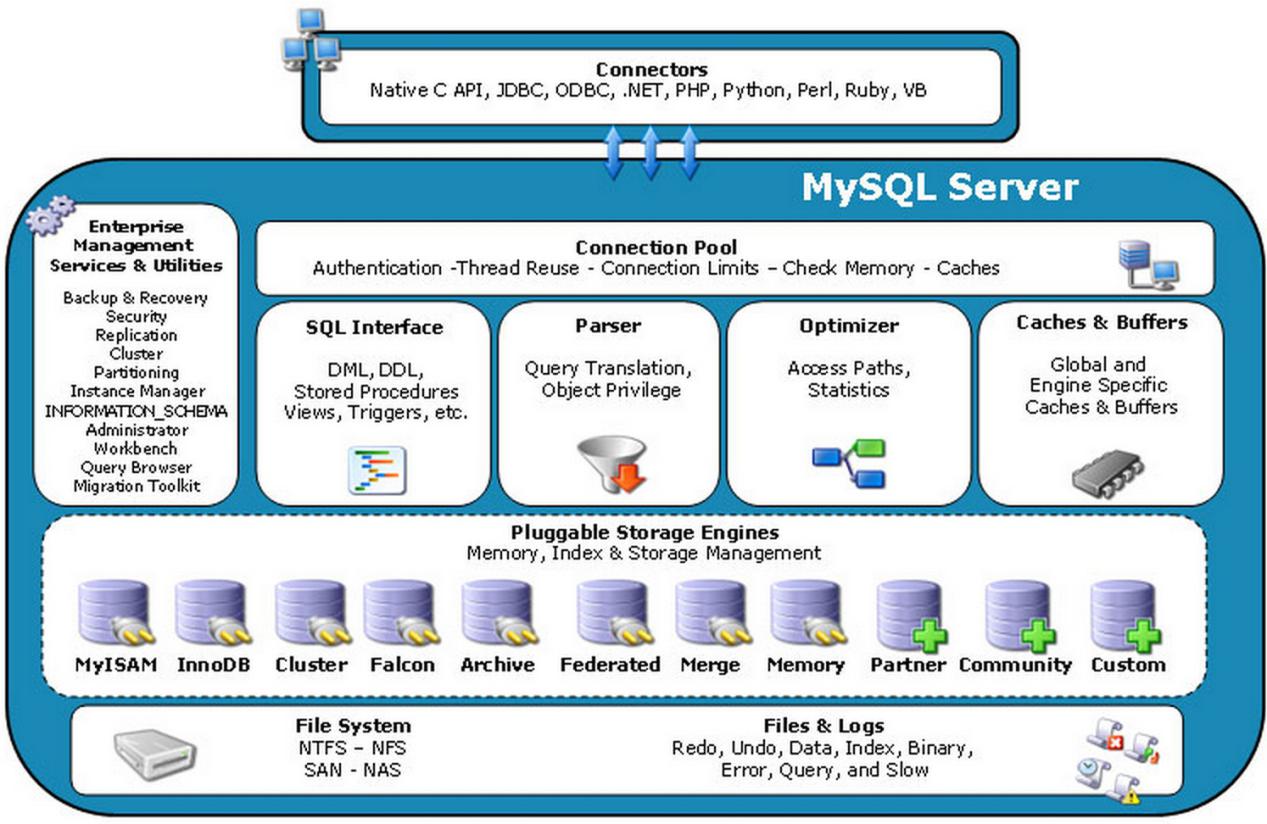
Sql全称：structured query language

现目前的RDBMS有那些？Mysql 、oracle 、db2 、sqlserver、marial DB()

现目前的NOSQL有哪些？(key-value形式存储)Mongodb、redis(更多使用内存)、Mem cache、hbase

Mysql的样子？(体系架构图) Mysql的重要组成部分？Sql接口、解析器、优化器、缓存、存储引擎等组件组成。



1 Connectors指的是不同语言中与SQL的交互

2 Management Serveices & Utilities： 系统管理和控制工具

3 Connection Pool: 连接池。

管理缓冲用户连接，线程处理等需要缓存的需求

4 SQL Interface: SQL接口。

接受用户的SQL命令，并且返回用户需要查询的结果。比如select from就是调用SQL Interface

5 Parser: 解析器。

SQL命令传递到解析器的时候会被解析器验证和解析。解析器是由Lex和YACC实现的，是一个很长的脚本。

主要功能：

a . 将SQL语句分解成[数据结构](http://www.yidianzixun.com/home?page=channel&keyword=%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84)，并将这个结构传递到后续步骤，以后SQL语句的传递和处理就是基于这个结构的

b. 如果在分解构成中遇到错误，那么就说明这个sql语句是不合理的

6 Optimizer: 查询优化器。

SQL语句在查询之前会使用查询优化器对查询进行优化。他使用的是“选取-投影-联接”策略进行查询。

用一个例子就可以理解： select [uid](http://www.yidianzixun.com/home?page=channel&keyword=uid),name from user where gender = 1;

这个select 查询先根据where 语句进行选取，而不是先将表全部查询出来以后再进行gender过滤

这个select查询先根据[uid](http://www.yidianzixun.com/home?page=channel&keyword=uid)和name进行属性投影，而不是将属性全部取出以后再进行过滤

将这两个查询条件联接起来生成最终查询结果

7 Cache和Buffer： 查询缓存。

如果查询缓存有命中的查询结果，查询语句就可以直接去查询缓存中取数据。

这个缓存机制是由一系列小缓存组成的。比如表缓存，记录缓存，key缓存，权限缓存等

8 Engine ：存储引擎。

存储引擎是MySql中具体的与文件打交道的子系统。也是Mysql最具有特色的一个地方。

Mysql的存储引擎是[插件](http://www.yidianzixun.com/home?page=channel&keyword=%E6%8F%92%E4%BB%B6)式的。它根据MySql AB公司提供的文件访问层的一个抽象接口来定制一种文件访问机制（这种访问机制就叫存储引擎）

现在有很多种存储引擎，各个存储引擎的优势各不一样，最常用的MyISAM、[InnoDB](http://www.yidianzixun.com/home?page=channel&keyword=InnoDB)、Heap(Memory)、NDB

mysql的存储引擎包括：MyISAM、InnoDB、BDB、MEMORY、MERGE、EXAMPLE、NDBCluster等，其中InnoDB和BDB提供事务安全表，其他存储引擎都是非事务安全表。

默认下My[Sql](http://www.yidianzixun.com/home?page=channel&keyword=Sql)是使用MyISAM引擎，它查询速度快，有较好的索引优化和数据压缩技术。但是它不支持事务。

[InnoDB](http://www.yidianzixun.com/home?page=channel&keyword=InnoDB)支持事务，并且提供行级的锁定，应用也相当广泛。

[Mysql](http://www.yidianzixun.com/home?page=channel&keyword=Mysql)也支持自己定制存储引擎，甚至一个库中不同的表使用不同的存储引擎，这些都是允许的。

最常使用的2种存储引擎:

1.Myisam是Mysql的默认存储引擎，当create创建新表时，未指定新表的存储引擎时，默认使用Myisam。每个MyISAM在磁盘上存储成三个文件。文件名都和表名相同，扩展名分别是.frm（存储表定义）、.MYD(MYData，存储数据)、.MYI(MYIndex，存储索引)。数据文件和索引文件可以放置在不同的目录，平均分布io，获得更快的速度。

2.InnoDB存储引擎提供了具有提交、回滚和崩溃恢复能力的事务安全。但是对比Myisam的存储引擎，InnoDB写的处理效率差一些并且会占用更多的磁盘空间以保留数据和索引。

需求：循环往一个表中插入数据：

Id name(“”+i) age(随机数)

CREATE TABLE IF NOT EXISTS USER(

id BIGINT(11) NOT NULL AUTO\_INCREMENT,

NAME VARCHAR(45) DEFAULT NULL,

age INT(1) DEFAULT 1,

PRIMARY KEY(id)

)

ENGINE=INNODB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8;

在mysql中的声明字段：

使用关键字：declare

普通类型申明格式如下：

Decalare 字段名 字段类型(位数) [default 默认值];

如:declare name varchar(45) default ‘’;

在mysql中的赋值：

使用关键字：set

如: Set i =100;

在mysql中判断常用的是if...end if 、if...else... 和 if...else if...else...等。

这几个的if else语句格式：(控制条件用可以and或者or)

If 控制条件 then

判断体; //每一句用分号结束，

End if;

If 控制条件 then

判断体; //每一句用分号结束

Else

判断体; //每一句用分号结束

End if;

If 控制条件 then

判断体; //每一句用分号结束

Elseif 控制条件 //注意Elseif不能分开

判断体; //每一句用分号结束

Else

判断体; //每一句用分号结束

End if;

在MySQL存储过程的语句中有三个标准的循环方式：WHILE循环，REPEAT循环以及LOOP循环，还有一种非标准的循环方式：GOTO，不过这种循环方式最好别用，很容易引起程序的混乱。其中最常见的是while和repreat

这几个循环语句的格式如下：

WHILE 控制条件 DO

循环体; //每一句用分号分隔开

END WHILE;

REPEAT

循环体; //每一句用分号分割开

UNTIL 控制条件 //不能加分号

END REPEAT;

循环的两个小例子：

While循环：

BEGIN

/\*

\*/

DECLARE count INT(11);

DECLARE i INT(11);

DECLARE userName VARCHAR(45);

DECLARE age INT(2);

SET count = 10;

SET i =0;

WHILE i < count DO

SET userName = CONCAT('1510A',i);

SET age = ROUND(ROUND(RAND(),2)\*100,2);

INSERT INTO `USER`(`NAME`,age) VALUES (userName,age);

SET i = i+1;

END WHILE;

END

Repeat的循环：

BEGIN

/\*

\*/

DECLARE count INT(11);

DECLARE i INT(11);

DECLARE userName VARCHAR(45);

DECLARE age INT(2);

SET count = 10;

SET i =0;

REPEAT

SET userName = CONCAT('1510A12',i);

SET age = ROUND(ROUND(RAND(),2)\*100,2);

INSERT INTO `USER`(`NAME`,age) VALUES (userName,age);

SET i = i+1;

UNTIL i > count

END REPEAT;

End

在mysql中的游标，游标相当于咱们编程中的动态数组。

游标类型申明格式：

Delcare 游标名称 cursor for

Select 被查询字段名 from 表名 where条件;

定义sql异常标记\_done ;

DECLARE CONTINUE HANDLER FOR [SQLEXCEPTION] SET t\_error = 1;

Open 游标名称;

Repeat

Fetch 游标名称 into 新字段名;

循环体;

Utile sql异常标记\_done

End repeat;

Close 游标名称;

在mysql中的事务：

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET t\_error = 1;

START TRANSACTION;

IF t\_error = 1 THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

在Mysql中的内部函数：

在mysql中的查询语句：

在mysql中的优化方式：