

# Mysql基础查询（一）

## 第三章

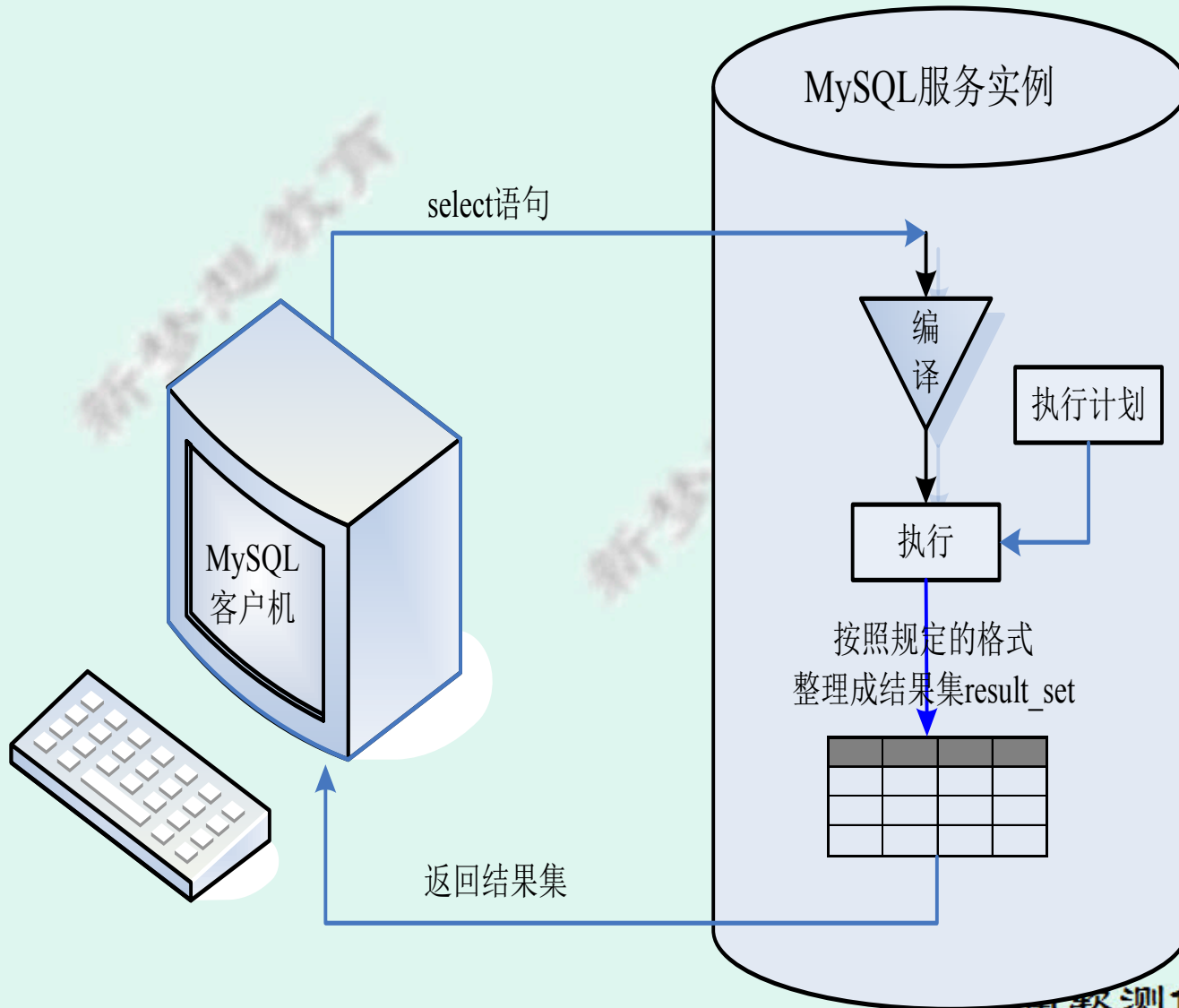
# 课程内容

- 1、基本的查询语句
- 2、排序操作---order by
- 3、limit, distinct关键字
- 4、模糊查询  
---between ...and\in\like\is null\逻辑表达式

# 课程目标

- 熟练掌握单表查询操作的相关语句

# select语句概述



# 1.1 查询语句

select语句的语法格式如下:

select 字段列表

from 数据源

[ where 条件表达式 ]

[ group by 分组字段

[ having 条件表达式 ]

]

[ order by 排序字段 [ asc | desc ] ]

## 1.2 查询表达式

- 每一个表达式表示想要的一列，必须至少有一列，多个列之间以逗号分隔
- \*表示所有列，tbl\_name.\*可以表示命名表的所有列
- 查询表达式可以使用[as] alias\_name为其赋予别名

# 1.3单表简单查询

- 查询表中全部数据

```
SELECT * FROM stu;
```

- 查询表中特定字段

```
SELECT id,newname FROM stu;
```

- 为字段取别名

```
SELECT username AS '姓名', sex '性别'  
FROM usertbl;
```

- 为表取别名

```
SELECT * FROM stu u;
```

# 单表简单查询

- 为字段指定表名

```
SELECT usertbl.username,usertbl.sex  
FROM usertbl;
```

- 为表名指定库名

```
SELECT username,sex,age FROM  
cms.usertbl;
```

- 使用常量进行查询

```
SELECT 1,2,3,username FROM usertbl;
```



## 2.1 Order By排序

- ORDER BY 语句用于根据指定的列对结果集进行排序。
- ORDER BY 语句默认按照升序对记录进行排序。
- 影响排序的关键字：ASC，DESC。

# 排序案例

- 按出生日期排序显示用户信息

```
SELECT * FROM users ORDER BY birth;
```

- 按工资排序降序显示用户信息，如果工资相同则按姓名字母顺序显示

```
SELECT * FROM users WHERE salary IS NOT NULL  
ORDER BY salary DESC, cname ASC;
```

## 3.1 Limit关键字

- 在我们使用查询语句的时候，经常要返回前几条或者中间某几行数据

```
SELECT * FROM table LIMIT [offset,] rows | rows
```

- LIMIT 子句可以被用于强制 SELECT 语句返回指定的记录数。
- LIMIT 接受一个或两个数字参数。参数必须是一个整数常量。
- 如果给定两个参数，第一个参数指定第一个返回记录行的偏移量，第二个参数指定返回记录行的最大数目。初始记录行的偏移量是 0(而不是 1):

## 3.2Limit实战

- 检查记录行，第0行开始，共返回10行记录

```
SELECT * FROM table LIMIT 0,10;
```

- 检查记录行，第100行开始到最后1行(一个很大的值)

```
SELECT * FROM table LIMIT 99,18446744073709551615
```

- 返回前5行记录

```
SELECT * FROM table LIMIT 5
```

## 3.3 distinct关键字

- 使用distinct过滤结果集中的重复记录  
使用方法如下：

Select distinct 字段名 from 表名；

例如：

查询出stu表中所有的姓名，去掉重得的姓名

```
SELECT distinct newname FROM stu;
```

# 4.1带条件的查询

- 带条件查询的基本语法

SELECT \*|字段名 FROM 表  
WHERE 条件表达式;

WHERE 条件	查询条件	符号
	比较	=、<、<=、>、>=、!=、<>、!>、!<、<=>
	指定范围	BETWEEN AND 、 NOT BETWEEN AND
	匹配字符	LIKE 、 NOT LIKE、 IN、 NOT IN
	是否为空	IS NULL 、 IS NOT NULL
	多个查询条件	AND 、 OR

## 4.2条件比较

- 查询姓名是zhangsan的员工信息

```
SELECT * FROM users WHERE cname='zhangsan';
```

- 查询id小于5的员工信息

```
SELECT * FROM users WHERE id<5;
```

- 查询工资不为5000的员工信息

```
SELECT * FROM users WHERE salary<>5000;
```

## 4.3指定范围查询-between-and

- 查询工资在5000~8000之间的员工信息

```
SELECT * FROM users  
WHERE salary BETWEEN 5000 AND 8000;
```

- 查询工资不在5000~8000之间的员工信息

```
SELECT * FROM users  
WHERE salary NOT BETWEEN 5000 AND 8000;
```

注意:

- 1.BETWEEN...AND...语句要求值小的在前面，值大的在后面
- 2.BETWEEN...AND...语句查询范围包含起始终止值
- 3.NOT BETWEEN...AND...语句查询范围不包含起始终止值



## 4.4指定范围查询---in

- 查询工资为5000， 6500 ， 8000的员工信息

```
SELECT * FROM users WHERE salary IN (5000,6500,8000);
```

- 查询姓名为： qianli， laoluo的员工信息

```
SELECT * FROM users WHERE cname IN (QianLI,LaoLuo);
```

- 查询工资不为5000， 8000的员工信息

```
SELECT * FROM users WHERE salary NOT IN (5000,8000);
```

思考：

BETWEEN...AND...查询与IN查询有何区别？

## 4.4 匹配字符查询---like

- 通配符

'\_' 号 表示一个字符

**A Like 'C\_'**

'%' 号 表示0个或多个字符

**B Like 'CO\_%'**

### 扩展的字符

“.” 匹配任何单个的字符。一个字符类

“ [ ] ” 匹配在方括号内的任意单个字符

“ \* ” 匹配零个或多个在它前面的东西

“ ^ ” 表示以什么开头的字符

“ \$ ” 表示以什么结尾的字符

使用扩展字符时，必须使用REGEXP和NOT REGEXP操作符（或RLIKE和NOT RLIKE）。

例：

1、匹配单个字符，查询出stu表中所有的id

```
select * from stu where id rlike '.';
```

2、查询出stu表中，以n开头的姓名

```
Select * from stu where newname rlike '^n';
```

3、查询出stu表中，查询出以a-z开头的姓名

```
Select * from stu where newname rlike '^[a-z]';
```

4、查询出stu表中，以n结尾的姓名

```
Select * from stu where newname rlike 'n$';
```

## 4.5匹配字符查询---like

- 查询姓zhang的员工信息

```
SELECT * FROM users WHERE cname LIKE 'zhang%';
```

- 使用%通配任意多个字符，使用\_通配1个字符

```
SELECT * FROM users WHERE cname LIKE 'zhangsa_';
```

- 使用NOT LIKE表示不满足条件的结果

```
SELECT * FROM users WHERE cname NOT LIKE 'zhang%';
```

# 4.6是否为空---IS NULL

- 查询工资为空的员工信息

```
SELECT * FROM users WHERE salary IS NULL;
```

- 查询工资不为空的员工信息

```
SELECT * FROM users WHERE salary IS NOT NULL;
```

备注：

查询工资为空，可以使用 $\lt;=\gt$ 表达式

```
SELECT * FROM users WHERE salary  $\lt;=\gt$  NULL;
```

## 4.7 多个查询条件-逻辑表达式

- 查询姓zhang的员工信息且性别为男的员工信息

```
SELECT * FROM users WHERE cname LIKE 'zhang%'  
AND sex = '男'
```

- 查询1990年以前出生的或者工资大于8000的员工信息

```
SELECT * FROM users WHERE birth < '1990-1-1' OR  
salary > 8000
```

# 多个查询条件-逻辑表达式

- 查询编号大于等于3年龄不为NULL的用户

```
SELECT * FROM users WHERE id >=3 AND age is NOT NULL
```

- 查询编号在5~10之间且用户名为单名（4位）的用户

```
SELECT * FROM users WHERE id BETWEEN 3 AND 10 AND  
cname LIKE '____'; (四个下划线)
```